Tether-free Driveline Control for Water Propulsion Devices

Isak Marklund, Axel Müller

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2023

MASTER THESIS

Tether-free driveline control for water propulsion devices

Axel Müller
ax2685mu-s@student.lu.se

Isak Marklund
is4484ma-s@student.lu.se

November 23, 2023

Master's thesis work carried out at
the Department of Design Science, Faculty of Engineering, Lund University.

Supervisors: Günter Alce, gunter.alce@design.lth.se
Dimitrios Triantafillidis, dimitris.triantafillidis@radinn.com

Examiner: Joakim Eriksson, joakim.eriksson@design.lth.se

Abstract

For many machines, safety requires the constant presence of an operator, with the risk of damage or danger if the operator is unintentionally absent. Dead man's switches (DMS) are commonly used to halt operations if this absence is detected, often relying on physical elements like leashes or buttons. However, these solutions can hinder operator convenience and equipment aesthetics. For devices with electrical brushless DC motors, motor data can be generated and analyzed. This opens the door for machine learning models to discern patterns signifying operator presence. The potential benefits are most evident in personal devices, where user activity directly influences motor data. This study delves into the feasibility of a software-based, tether-free, DMS for a jet surfboard.

Initially, user opinions on the existing leash-and-magnet DMS were gathered, assessing its impact on user experience through surveys and interviews. Results showed that a tether-free DMS solution would improve user experience mainly in regards to convenience and comfort. The existing DMS is regarded as simple, safe and reliable - characteristics that would need to be reflected by a tether-free DMS.

Subsequently, machine learning models were constructed to analyze ride log data in order to detect surfer presence. Two approaches were explored, detecting when someone is falling off the board, and detecting when the board is running with no one on it. It was found that strong negative and positive trends in the motor data (corresponding to deceleration and acceleration) were the characteristics that the models mostly identified with fall-off and off-board data, which resulted in some misclassifications. Results show that there are some possible differentiating features between different activities on board in some cases, but the models ultimately fall short of the performance requirements for a DMS.

Keywords: Dead man's switch, electric surfboard, jet surfboard, machine learning, user experience, usability testing, neural network, decision tree, bootstrap aggregating, time series

Sammanfattning

Det här arbetet undersöker möjligheten att implementera ett mjukvarubaserat dödmansgrepp för en jet-surfbräda. Ett dödmansgrepp (eng. dead man's switch, *DMS*) är en säkerhetsanordning som används i maskineri och fordon som kräver avstängning när förare eller användare inte är närvarande. Gemensamt mellan dessa fordon och maskiner är att det finns risker för omgivning och användare om användaren av någon anledning inte har möjlighet att kontrollera dem. Motorbåtar och vattenskotrar är exempel på fordon som nyttjar dödmansgrepp: en rem kopplas mellan föraren och tändningen så att motorn stängs av om föraren hamnar i vattnet. Att ha en rem fäst vid sig kan dock vara obekvämt för föraren.

Med elektriska borstlösa likströmsmotorer är det möjligt att samla och analysera data om motoranvändningen. Det öppnar möjlighet för att med hjälp av maskininlärningsmodellering upptäcka mönster i datan som tyder på att föraren är närvarande eller frånvarande. Den här typen av analys är mest lovande för mindre fordon, som jet-surfbrädor, då förarens vikt och rörelser har mer direkt påverkan på styrningen och farten av fordonet.

I den första delen av arbetet genomfördes en användarstudie där åsikter gällande det nuvarande dödmansgreppet, som använder en rem kopplad till en magnet som fästs i brädan, undersöktes. Användarstudien bestod av intervjuer och en enkät. Resultaten visade att en DMS-lösning utan rem och magnet hade kunnat vara en förbättring ur komfort- och bekvämlighetssynpunkt. Det nuvarande dödmansgreppet anses dock vara enkel, säker och pålitlig – viktiga egenskaper som också skulle behöva återfinnas i ett alternativ som inte använder rem och magnet.

I den andra delen så konstruerades maskininlärningsmodeller för att analysera motordata från riktiga åk som loggats. Två sätt att detektera användarnärvaro utforskades under arbetet: att detektera avramlingar och att detektera när brädan körs utan någon på (*"off-board"*). Det framkom att det som modellerna främst klassifierade som avramlingar eller "off-board" var vid starka inbromsningar och snabb acceleration, vilket gav upphov till en del felklassificeringar. Resultaten visade att det finns möjliga skillnader mellan olika aktiviteter på brädan, men även många likheter, som leder till att modellerna i slutändan inte uppnår de prestandakrav som ställs av en DMS.

Keywords: Dödmansgrepp, elektrisk surfbräda, jetdriven surfbräda, maskininlärning, användarupplevelse, användbarhetstestning, neurala nätverk, beslutsträd, bootstrap-aggregering, tidsserier

Acknowledgements

We'd like to thank our thesis supervisor from LTH, Günter Alce, for his feedback and support. We're also thankful to Radinn for allowing us to conduct our thesis at their company and for their assistance with equipment and technical support. Special thanks to our company supervisor, Dimitrios Triantafillidis, and Radinn's head of platform, Daniel Nordahl. We also appreciate those who participated in our user tests and interviews. Finally, a thank you to our friends and family for their continuous support.

Contents

Chapter 1

Introduction

1.1 Background

The operation of any type of water vehicle presents the risk of the operator inadvertently falling into the water. If a rider falls off, the inertia of movement can carry the vehicle far away from the operator before stopping completely. This is an annoyance as the operator needs to swim back to the board in order to climb back on, which in bad weather conditions also could be a risk to the operator's health and safety as the swim becomes more difficult. Staying in balance is an active challenge for a surfer, as surfboards are light and responsive to the conditions on the water and the operators usually are not able to hold on to anything for support while standing.

For motorized water vehicles, the greater speed and weight can cause risk to both operator and surroundings, requiring a fail-safe mechanism to shut off the engine when the operator is not present or incapacitated, commonly referred to as "dead man's switch" (DMS) [18]. Personal watercrafts and outboard motors are usually equipped with a DMS where a cord, leash or lanyard needs be attached to the ignition of the engine in order for it to run. The leash should then be attached to the operator, so if they are thrown away from the motor or vehicle, the leash will release from the ignition, triggering the dead man's switch and shutting down the engine [27].

Radinn is a company that produces electric motorized surfboards [25]. Currently, the electric surfboards at Radinn are equipped with a DMS mechanism similar to the ones used by outboard motors and jet skis. To operate the electric surf board, a magnet needs to be attached to the board. This magnet is connected to a cord, which wraps around the operator's ankle. If the operator is thrown off, the cord pulls on the magnet until it is detached, shutting down the motor completely.

This work intended to explore the possibilities and benefits of using machine learning-based pattern recognition to implement a DMS mechanism without a leash. An early analysis of data from the existing ride logs was conducted in order to find possible patterns in the data and to determine the feasibility of using this data for DMS. A user study was then conducted in order to gauge the interest in this feature. Then, by utilising machine learning (ML) models to categorize data from the ride logs, the ambition was for the models to be able to discern when the rider is present on board.

1.2 Purpose and research questions

The main goal of this thesis is to investigate how the user experience may be improved by implementing a tether-free DMS, meaning a DMS function without the need to use the leash. Machine learning models will be constructed to attempt to identify when nobody is present on the board, based on data from the motor. Furthermore, the goal is to evaluate the models in regards to their robustness and precision. The following research questions are intended to reflect these goals:

- **Research question 1:** How is the user experience affected by the current DMS system?
- **Research question 2:** What are the motivations for implementing a tether-free DMS system?
- **Research question 3:** How well can rider presence be detected with machine learning, utilizing data from the ride logs?

1.3 Limitations

The research conducted in this thesis was influenced by several limitations that should be considered when interpreting the result and the subsequent discussion. Natural limitations for master thesis works are time and scope. One semester of time, approximately 20 weeks, is available in total, requiring a focused scope to achieve a significant result.

Due to time constraints, no DMS algorithm was implemented on an actual surfboard. An implemented algorithm would be beneficial for testing purposes and for demonstrating the result, but is unfortunately beyond the scope of the thesis. In addition to transferring any machine learning model to the programming languages required to run them on an embedded system, another step of testing would have to be added in order for it to be fruitful for the thesis. This would be too time consuming.

As a result of not implementing the feature on a board, the design aspect of the thesis becomes limited. Further inquiries could otherwise be made into the design aspect of the tether-free DMS feature, such as exploring different ways of indicating when the board is usable. Another iteration of user centered design would be beneficial for the rigidity of the thesis as a whole.

The absence of annotated pre-existing data poses a challenge in training and validating the machine learning models. Annotated data that differentiates between someone falling off the board and someone hopping off the board deliberately would be beneficial for training, as the fall-offs could be trained on specifically to increase accuracy. This is a natural limitation that has directly influenced the need for the work in this thesis. For this thesis, differentiation between fall-offs and deliberate hop-offs in the ride logs needs to be done through scripting, which in itself is limited by the quality of the data, knowledge about the data, and the available tools.

Furthermore, incorporating additional sensor inputs could potentially improve the likelihood of successfully executing the assigned tasks; however, it was specified by Radinn that only eRPM, motor current and DMS data were utilized in the thesis.

1.4 Sustainable development goals

Making way for the development of a safer alternative to the leash-based DMS switch can inhibit accidents and injuries, thus enhancing the well-being and health of individuals utilizing personal watercrafts. This initiative is in line with goal number three, to "ensure healthy lives and promote well-being for all at all ages," from the Sustainable Development Goals established by the 2030 Agenda for Sustainable Development [20](see Figure 1.1). The goal is to enhance participant safety and well-being in water sports by promoting the use of safer technologies.

Figure 1.1: The 12 Sustainable development goals by United Nations towards a sustainable future [20].

1.5 Related work

The topic of tether-free presence detection on personal watercraft is sparsely researched. Frotan and Moths [6] evaluated the use of a pulse-coherent radar module as a way of implementing a tether-free presence detection system for Radinn boards. Provided a protective dome, a radome, was placed around the radar, it was found that it is possible to detect blockage of the radar (implying presence of a surfer) by strictly observing the direct leakage of the signal.

A research topic that is related to tether-free presence detection is driver profiling, identifying the characteric driving styles of different people using only data generated by the cars that were driven. Many works exist concerning driver profiles, some of which make use of data that is strictly from the on-board diagnostics (OBD), using that data to derive information about drivers. These works provided a foundation for the machine learning model aspect of this thesis, by way of showing different classifiers that can work for time series data, and in selecting or deriving features from the data.

Kumar and Jain [11] used supervised learning algorithms on OBD data to classify drivers according to driver styles. They defined key driving events, such as high-speed braking and rapid acceleration, that were used for the classifications. The models performed well, with 99%, 99%, and 100%, accuracy for Support Vector Machine, AdaBoost, and Random Forest, respectively. This shows that there may exist patterns in motor data that differentiate between users, which could be applied to finding anomalies in riding data that represent someone falling off.

Kwak et al [12] proposed an anti-theft mechanism that utilized machine learning based driver profiling to detect when a driver did not match preexisting profiles. They used 60 second time windows of 15 features sampled at 1 Hz, utilising a Random Forest model to classify drivers. The model performed at 99 % accuracy. This level of performance at a relatively low sample rate shows that low frequency data can exhibit unique characteristics that enable classification.

Plenty of work has also been done on human activity recognition, the task of identifying movement and gait in humans, based on sensor data such as accelerometers. Lee et al [13] proposed a convolutional neural network-based approach that achieved 92.71 % accuracy in recognizing walking, running, and staying still, using 1 Hz accelerometer data that had been condensed from three axes into one dimensional data. Similarly to [12], this shows promise in classifying activities based on relatively low frequency data, as well as using only one-dimensional data to do so.

The work of this thesis differs from the previously mentioned works in that there is a single type of event that needs to be identified from the data. However, the event that is to be identified, falling off the board or attempting to accelerate when no one is on board, is an event that lasts for only a few seconds at most. There is a need for a strong classification algorithm, good feature extraction, and/or high quality data, in order for this type of event to be properly classified.

Chapter 2

Theory

This chapter will outline the theoretical background for all aspects of this thesis. Section 2.1 will outline what a personal watercraft is and how it operates, how brushless motors work, concluding with a description of the Radinn jet surfboard. Section 2.2 will outline the design process and the underlying theory for the user tests that were conducted. Section 2.3 details the machine learning theory that is utilized for analyzing the motor data.

2.1 Personal watercrafts

A personal watercraft (PWC) is typically a small and lightweight watercraft with high acceleration and maneuvering capacity designed to carry approximately 1-2 individuals. The characteristic handling and maneuvering of a PWC is strongly connected with the water-jet propulsion system. The PWC's unique feature characteristics make these watercraft excellent at operating in diverse water conditions. Traditionally there have been two main types recognized: the stand- up type and the sit-down type, commonly referred to as "Jet-Ski" (see Figure 2.1). Despite their shared name, these two types differ in terms of operational methods and design features. The stand-up type is designed for a more extreme and thrilling ride for a single person. To operate this type of PWC greater expertise and physical capabilities is required of the rider. The sit-down model is generally bigger and designed for a more comfortable and beginner friendly ride, therefore these models are popular for sightseeing and recreational joyrides. These sit-down models are equipped with larger hulls and engines, allowing them to engage in various water sports activities, including towing water skis and tubes.

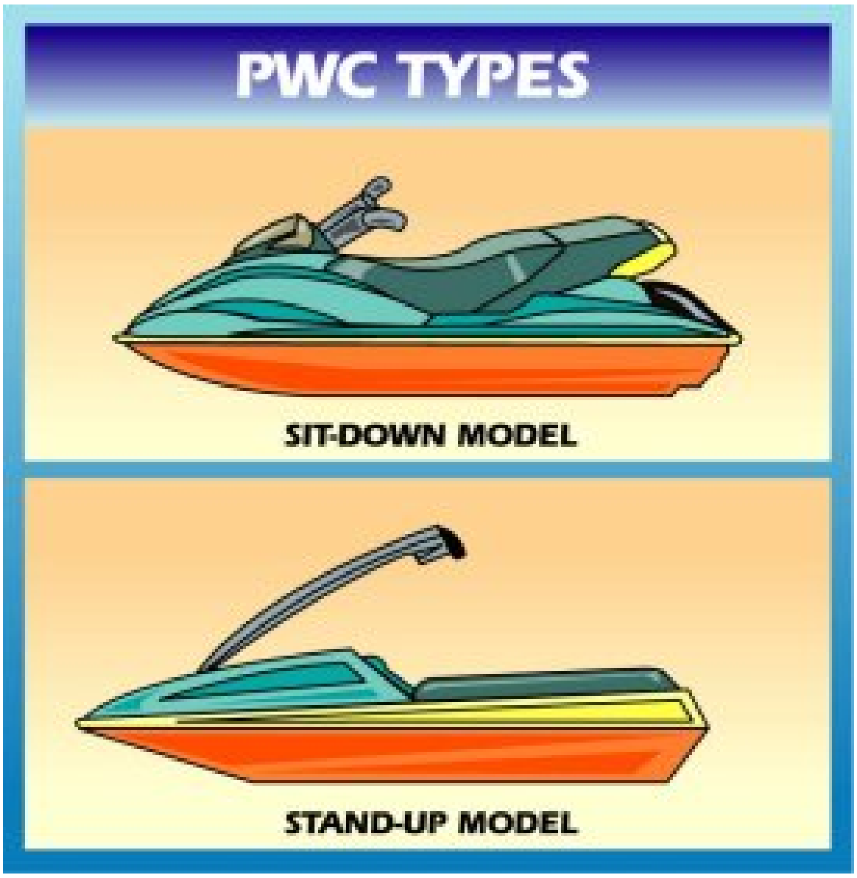

Figure 2.1: Typical types of PWC [5].

Additionally, apart from being used for recreational purposes, Jet-Skis are also utilized professionally by a wide range of official agencies, organizations, and companies. These versatile watercrafts find practical applications in various fields such as law enforcement, search and rescue operations, emergency medical services, coastal patrols, and environmental monitoring. The maneuverability and speed of Jet-Skis make them valuable assets for professionals requiring swift and agile waterborne transportation[5]. Injuries associated with Jet-Skis are collisions with other watercrafts or stationary objects, falling off the craft at high speeds, and the exposure of high pressure water-jet to the body [14].

2.1.1 Brushless motors

A brushless DC motor (BLDC), also referred to as an electronically commutated motor, is an electric motor that utilizes an electronic controller to regulate the electrical fields within the motor and induce rotational motion. This type of motor exhibits a lengthy operational lifespan, good durability, and low maintenance requirements. In contrast, a brushed DC motor relies on brushes to establish and sustain contact with the commutator, allowing the passage of electrical current into the rotor windings. Such motors need more frequent maintenance, including periodic brush replacements, and are generally less durable [29]. See figure 2.2 for a comparative diagram of these two types of motors.

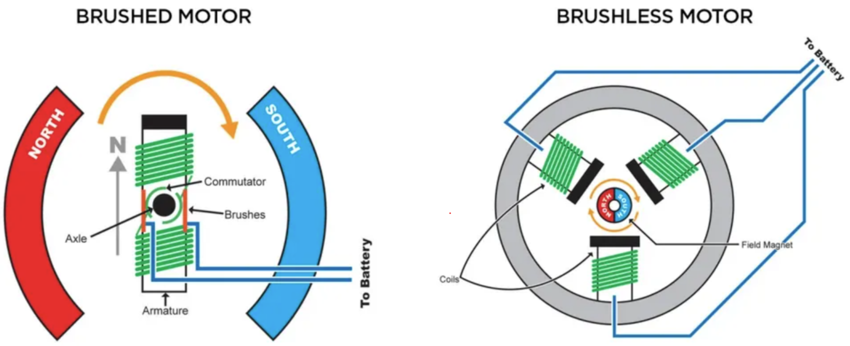

Figure 2.2: Internal wiring differences between conventional DC motors and BLDC motors [15].

In general, when the amount of electric current supplied to a BLDC motor is increased, the strength of the magnetic field produced by the motor's magnets also increases. This stronger magnetic field leads to a proportional increase in the motor's torque, which induce higher rotational force. Assuming the motor is operating under a constant load, this increased torque results in a higher rotational speed, typically measured in RPM (Revolutions Per Minute). In this context, load refers to the external mechanical resistance, the force that opposes the motion or operation of the motor. Mathematically, the relationship between torque, power, and speed can be expressed as follows:

$$\text{Torque (Nm)} = 9.5488 \times \frac{\text{Power (kW)}}{\text{Speed(RPM)}} \tag{2.1}$$

Hence, if current is maintained at a constant level and the load of the engine is decreased, rotational speed (RPM) of the rotor will increase proportionally. This relationship between current, load, and RPM is because of the torque-speed characteristics of the motor. When the load on the motor is reduced, there is less resistance for the motor. As a result, the motor can rotate more freely, leading to an increase in rotational speed (RPM). However, it's important to note that this relationship holds true only when the current supplied to the motor remains constant. By keeping the current constant, the magnetic power generated by the motor's magnets remains the same. This means that the torque produced by the motor remains constant as well. Hence, the rotation of the motor will increase. Therefore, if the load on an engine decreases but the current supplied to the motor stays the same, the RPM of the motor will increase proportionally [8].

2.1.2 Water jet propulsion

A watercraft equipped with water jet propulsion system utilizes the principle of reactive force by sending pressurized water from a chamber to propel the vessel in the opposite direction. The process involves the motor-driven internal propeller pressurizing the water within the chamber, which exits through the nozzle, resulting in the propulsion of the watercraft [1].

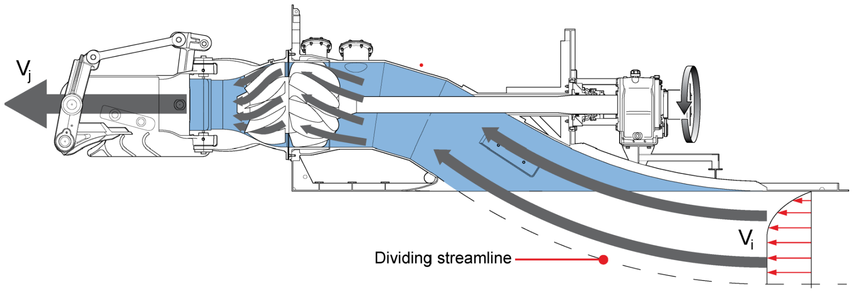

Figure 2.3: Water flow in water jet propulsion [1].

The utilization of water-jet propulsion removes the safety hazard accruing in more conventional propulsion systems, such as those with an externally exposed propeller. By moving the propeller inside a chamber in the hull and therefore relying only on the jet nozzle and inlet as external components, the risks for personal injury associated with touching moving parts are removed. However, the powerful jet stream produced by the water jet propulsion system can still be a danger due to the high-pressure water being expelled from the nozzle [17]. Furthermore, water jet propulsion systems heavily reduce water draft, referring to the minimum depth of water necessary for the watercraft to operate. By relocating the nozzle to the back of the vessel and submerging it underwater, a water jet watercraft can operate in shallower water than the alternatives. Water-jet propulsion systems gives significant advantages to watercrafts in specific usage areas such as patrol and rescue operations, recreational rides, and water sports activities. They offer good maneuverability, making navigation through narrow channels and debris-filled areas more efficient. Additionally, the absence of exposed propellers enhances safety for divers and individuals in the water[5].

2.1.3 Radinn's X-Sport jet surfboard

The Radinn X-Sport jet surfboard consists of three main components. First, there's the board itself, often referred to as the hull (see Figure 2.4a). Next is the jetpack, the waterjet component (see Figure 2.4b). This part houses the motor and all the technical elements that power the board. Lastly, there is the lithium-ion battery that supplies power to the jetpack (see Figure 2.4c). To use the board, additional items are necessary, including a wireless hand controller for throttle control and monitoring the battery level (see Figure 2.4e).

The magnetically connected board leash (see Figure 2.5 and 2.4d) acts as a safety DMS feature, by turning off the throttle when the rider gets separated from the board. The leash is attached to the leg of the surfer with a velcro strap, and the magnet is connected to the board in order for the board to become usable. When the surfer becomes separated from the board, the leash is tugged by the leg, and the magnet is disconnected from the board, disabling the driveline.

The jetpack collects real-time data logs capturing details from both the battery and the motor during a ride. These logs are saved in the board's internal memory. Upon connecting the board to a phone though the Radinn app, these logs are uploaded to the cloud (see Figure 2.6). The recorded data contains a variety of metrics, such as the current within the motor, information about the motor (such as the impeller's rotational speed), GPS data, and more.

(a) X-Sport board

(b) Jet-pack

(c) Battery

(d) Board leash

(e) Hand controller

Figure 2.4: Components of the X-Sport jet surfboard. All images sourced from Radinn's website [25].

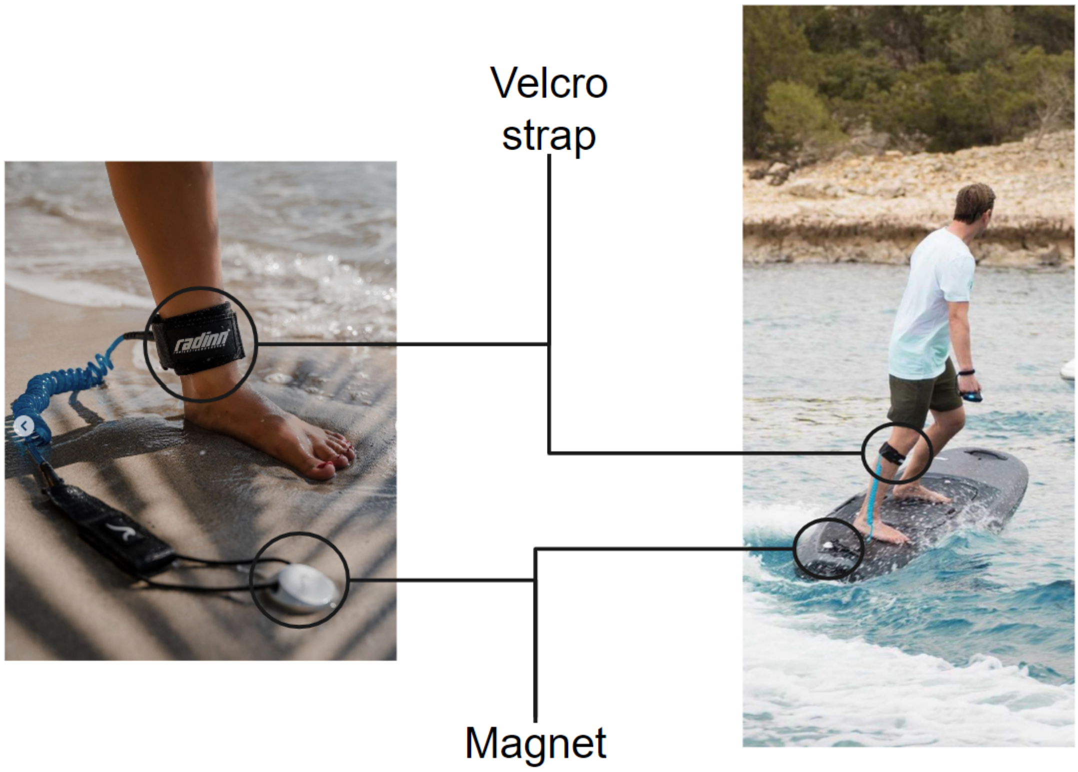

Figure 2.5: The leash consists of a coiled cord attached to the rider's leg with a Velcro strap at one end and a magnet attached to the board at the other end.

Figure 2.6: Illustration of logs uploaded from the board's internal memory and transfer the cloud via the Radinn app

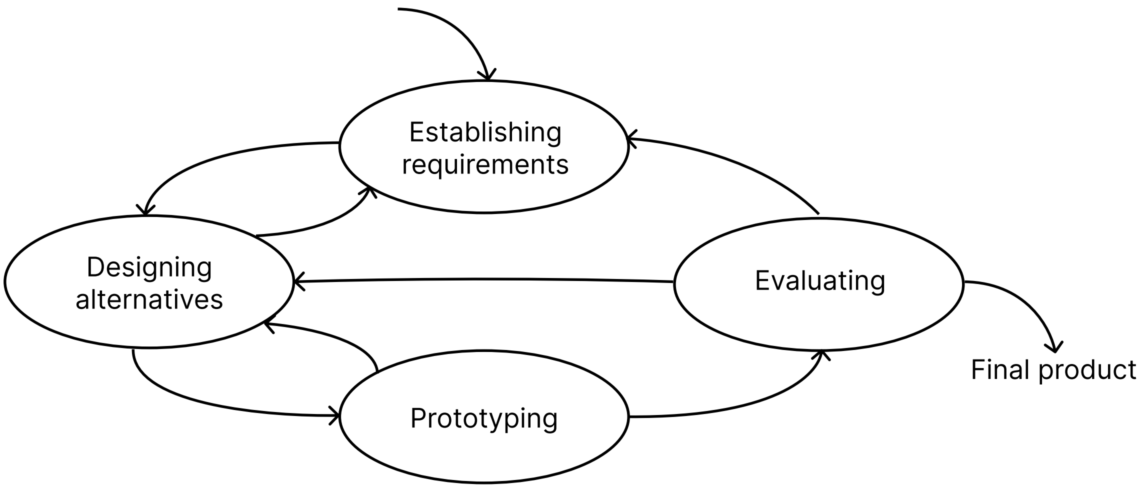

Figure 2.7: The typical iterative design process [23].

2.2 Design process

This section will present some basic theory regarding interaction design. The central concepts of user experience and user-centered design are presented. Following that, the methodology of conducting a user study is presented, from selecting participants, to gathering user data through surveys and interview and lastly, to assess the data from the study.

2.2.1 User experience

User Experience (UX) refers to how people feel and think when using a product, system, or feature[23]. It is about how users perceive and experience what they interact with. One important aspect of UX is usability, which describes how easy it is for users to accomplish their goals with a product or system. By understanding the user's experience through UX testing, developers and designers can make sure that the product works well and meets users' needs. They can improve its usability, making it easier and more enjoyable to use. This leads to happier users, increased engagement, and greater success for the product.

2.2.2 User-centered design

A user-centered design approach entails developing a product based on decisions that are entirely derived from evaluating the user feedback in every stage of the design process. This means that the user's thoughts and feelings regarding the product are central components, guiding the design decisions at each step. By actively involving users and incorporating their insights, the resulting product is likely to meet their expectations and improve their overall satisfaction of the product [23]. The process is typically iterative, continuously evaluating the design on users throughout the process. See Figure 2.7 for a diagram of a typical design process.

2.2.3 User experience metrics

When it comes to UX, metrics differ from conventional measurements typically employed in other tests. Instead of solely relying on objective metrics like length, height, or time, the focus is on capturing subjective user perceptions, emotions, and satisfaction levels. Additionally, data related to how well a test participant achieved their goals, such as time spent on specific tasks or deviations in navigation, can also provide valuable insights [23].

2.2.4 Data collection

This section explains quantitative and qualitative research methods, investigating their distinct approaches and applications. This section also explains how choosing the appropriate data gathering methods depends on the goals of the study. Some self-reported metrics are described, including Likert and Semantic Differential Scales, and the nuances of conducting interviews.

Quantitative vs. qualitative data

A quantitative study collects numbers to analyze and test ideas, while qualitative research provides detailed descriptions using words. Quantitative studies need a greater amount of participants than qualitative research. The proper approach depends on the goal of the study. If you want to check a theory or hypothesis, quantitative data is the more suitable option. For example, if a building owner wants to know how happy tenants are, they can ask everyone to rate different parts of the building. This gives statistical info. If they want more in-depth data, at the cost of fewer respondents, qualitative is better. They can ask a few tenants open-ended questions about what they like or dislike. This gives more detailed info and new insights, while also opening up the possibility of finding out user sentiments that were not specifically covered by the questions. Another option is to mix both approaches. Start with qualitative to get detailed insights, then do a bigger quantitative study to see the bigger picture. This way, you get some of the advantages of both kinds of studies [28].

Gathering data from users

To gather relevant information during a user test, various methods such as interviews or forms can be used. The choice of method depends on factors like the study's nature (quantitative or qualitative), the product being tested, user accessibility, and motivation. When gathering data through questionnaires or interviews, there are two types of questions: open-ended and close-ended. Close-ended questions provide predefined answer options, making responses easier and increasing the number of data points. However, they may limit the level of detail and make it challenging to capture subtle differences in responses. Open-ended questions allow for detailed responses but can result in irrelevant or repetitive information. Combining both types of questions can simplify evaluation and ensure comprehensive data collection. Open-ended questions are valuable in interviews, encouraging detailed and comprehensive answers. In semi-structured interviews, including open-ended questions is crucial. They let interviewees express thoughts deeply, leading to comprehensive answers. But questions should also allow quick assessment of attitudes or opinions [4].

Surveys

By measuring user satisfaction, usability, engagement and behavior, informed decisions can be made to improve the user experience. This approach helps identify areas for improvement, optimize design elements, and create more user-friendly designs. Self-reported metrics involve collecting information from users by asking about their experiences, preferences, and opinions. Three common methods for data gathering include open-ended questions and close-ended questions with answer alternatives on the Likert scale, or on the Semantic Differential Scale [30]. The Likert scale presents statements and asks users to choose from response options ranging from "strongly disagree" to "strongly agree." This structured approach measures attitudes and opinions by assessing the level of agreement or disagreement (see Table 2.1).

Table 2.1: Likert Scale

Scale	Description
1	Strongly disagree
2	Disagree
3	Neither agree nor disagree
4	Agree
5	Strongly agree

Semantic Differential Scales, similar to Likert scales, capture users' perceptions and emotions. Instead of fixed words, users rate their alignment between two opposite words on a scale (see Table 2.2).

Table 2.2: Semantic Differential Scales

Weak	o	o	o	o	o	o	Strong
Ugly	o	o	o	o	o	o	Beautiful
Cool	o	o	o	o	o	o	Warm
Amateur	o	o	o	o	o	o	Professional

Interviews

The way interviews are conducted can vary based on the specific goals, type, and circumstances of the study. One of the benefits of an interview is that it allows the interviewee to speak freely and provide their own answers. To take advantage of this benefit, it is important to design the questions in a way that allows for such open responses. In a semi-structured interview, it is crucial to include open-ended questions that provide the interviewee with the opportunity to express their thoughts and opinions in depth. However, these questions should still be designed in a way that allows for a quick assessment of the interviewee's attitude or opinion [24].

2.2.5 Evaluation of results

Due to the different methods of gathering quantitative and qualitative data, the evaluation methods also differ. Quantitative data analysis uses visualizations like charts and graphs to simplify complex data and relies on metrics like averages and variability for insights, particularly in data set comparisons. Qualitative interview data is often analyzed by identifying similarities in statements, and categorizing them into patterns and themes, for example by using a so-called affinity diagram.

Quantitative data analysis

When interpreting results derived from quantitative data collection, the use of visualizations through charts and graphs works as a useful methodology to effectively present extensive data sets in a simple way. Furthermore, key metrics like averages and measures of variability provide simple and powerful insights into the dataset, increasing the understanding of the underlying information. Visual representations of data are particularly strong when comparing two distinct groups or datasets, as they show a clear and insightful comparison [23].

Qualitative data analysis

By examining statements and segments of an interview independently, a clearer picture can be formed. When this process is repeated across multiple interviewees, statements and sections tend to show similarities. While they might not be identical, they often share resemblances. As a result, these resemblances can be grouped into categories, patterns and themes. A useful technique for analyzing qualitative data involves the creation of a so-called affinity diagram. This method applies to information gathered through qualitative means like in-depth interviews [10, 23].

2.2.6 Intention of study

When planning a UX test, understanding its purpose is crucial. Understanding the purpose helps set study objectives and gain insights into user interactions with the product, including frequency of use and reasons for choosing it. Here are examples of questions to consider [30]:

- Are you aiming to ensure that a new functionality provides an optimal user experience, or are you comparing the user experience of an existing product with certain standards?
- Do you want to identify the most significant usability issues or gather data on user preferences regarding the design?
- Are users primarily focused on completing a specific task and then discontinuing product usage, or will they be using the product repeatedly on a daily basis?

By exploring these questions, three key areas of interest emerge: user performance and preferences. User performance measures how effectively tasks are completed, considering factors like time taken, attempts, alternative paths, and overall success. Evaluating user preferences can be challenging due to diverse tastes, such as varying preferences for instructions. Some prefer detailed guidance, while others appreciate more subtle explanations.

2.2.7 User research methods and tools

User studies can be conducted in two main ways: moderated and unmoderated. In moderated tests, the conductor interacts directly with the user, guiding them through tasks and collecting information. The user's thoughts and observations are encouraged during the session. Moderated tests usually involve 8-12 participants and focus on collecting metrics related to issues such as frequency, type, and severity. However, self-reported metrics in these tests may have limitations, including the potential for over-generalization [28]. On the other hand, unmoderated tests provide more unbiased information as participants act without external influence. These tests are easier to conduct with larger groups, allowing for diverse user feedback. Participants in unmoderated tests may follow a predefined script or a few instructions, and their actions are often monitored using software for tracking interactions [23]. Both approaches have their advantages, but it is important to assess the questions asked and consider the limitations of self-reported data in moderated tests to ensure accurate and reliable results.

2.2.8 Participant selection

Recruiting participants with diverse backgrounds provides a wide range of perspectives and enhances product evaluation. To focus evaluations on specific factors, dividing participants into subcategories can be beneficial, leading to more precise metrics [26]. Inclusive testing is crucial for products targeting a broad audience, like microwaves, mobile ticket apps, or medical devices. Including diverse participants ensures usability for different users and gathers insights from various perspectives. However, too much diversity can hinder reliable results, especially for products designed for specific user groups. Careful participant selection is essential to avoid misleading outcomes. Nonetheless, when thoughtfully categorized, a large and diverse group of participants can provide valuable insights into the impact of design changes on different user groups [30].

2.3 Machine learning

Machine learning (ML) is an umbrella term for algorithms and data structures (referred to as models) that are aimed at learning structures within datasets. ML is a vast topic and there exists a number of models used for different kinds of ML. This section will explain some of the basic underlying theory and vocabulary of ML, along with the models that are used throughout this thesis.

2.3.1 Different types of machine learning

ML models can be divided up into two distinct types, unsupervised and supervised. For this thesis, unsupervised learning refers to clustering, where a model attempts to find discrete categorizations in *unlabeled* data. Supervised learning in this thesis is classification, where the model uses *labeled* data in order to learn the rules of an established categorization scheme, to classify new data [19]. In the scope of this thesis, supervised ML is used for identifying an activity (riding the board or falling, for example) based on motor data, while unsupervised is used in order to fine tune the definitions of some activities.

2.3.2 General machine learning concepts

This section will focus on some of the vocabulary and concepts that applies to ML as a whole. Most of this section will apply only to supervised learning, but some aspects, such as feature selection, also play a part in unsupervised learning.

Feature selection

The data that is passed as input to a ML model consists of a number of features. In time series data, a feature could be statistical metrics such as mean value and variance, or it could be the actual "raw" time series values themselves. Not all features are valuable to the classification task however, and features that are individually valuable might not combine to make the best and most efficient classification [22]. There might be redundancy between two features that makes a model less efficient, adding a redundant feature will only add calculations and complexity without strengthening the classifier.

Minimum Redundancy Maximum Relevance (mRMR) is a measurement used to calculate feature importance. It is a method of selecting features that prioritizes informative features while reducing redundancy. The method uses mutual information, an information theory metric which measures how much can be learned by one variable through another variable. For more information on mRMR, see [22].

Pearson's correlation coefficient (eq. 2.2) is utilized in this method to find linear relationship between a feature's output in relation to a data set of cluster of data points. In mRMR, Pearson's correlation coefficient is used to find features with robust linear relations. Higher coefficient values translates strong linear correlations, while values near 0 indicate weak correlations.

$$\text{Pearson Correlation}(X, Y) = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}} \tag{2.2}$$

Where:

$$\begin{aligned}
X &: \text{Dataset } X \\
Y &: \text{Dataset } Y \\
X_i &: \text{Individual data points in the dataset } X \\
Y_i &: \text{Individual data points in the dataset } Y \\
\bar{X} &: \text{Mean (average) of } X \\
\bar{Y} &: \text{Mean (average) of } Y
\end{aligned}$$

Datasets

In order for ML models to actually learn, they need to be provided with a dataset to learn from. This means a suitably large amount of data that is properly formatted to serve as input to the model, and in the case of supervised learning, categorized and annotated properly. A large subset of this dataset will serve as the *training set*, the data used for training the model. The training data is passed through the model, and the resulting classifications are used to adjust the model's parameters. A smaller subset serves as the *validation set*, that is used in between each training iteration to calculate the performance by passing it through the model and calculating the accuracy of the classifications [19]. Finally, a subset of data is reserved for the final performance calculation, this is called the *test set*. For unsupervised clustering models only one dataset is needed, since the model does not need to validate itself, but is supposed to "explore" the structure of data.

Objective functions and loss

The main idea behind the training of a model is to minimize the discrepancy between the model and the real world. To minimize the discrepancy is the same as optimizing the model's *objective function*. The purpose of the objective function is always the same, but could be formulated as to maximize the amount of correct predictions or to minimize the amount of errors.

Loss functions are commonly used objective functions. They compute the approximate "cost" of using the current model. In order to optimize the model, this function is to be minimized. Different types of loss functions are used depending on if the task is regression analysis or a classification problem. For classification, the loss metric used is commonly the *cross entropy*. This measures the dissimilarity between two probability distributions. In the context of loss, these are the probability distributions of the predictions and the actual observed value [2].

Data regularization, over- and underfitting

Real world data is generally noisy, complex and difficult to accurately extract features from. Due to this noisiness and complexity, it is likely that there is some overlap between the distributions of different categories of data. Overfitting a model means that the noisiness of the data is given too much weight by the model. Differences that result due to noisiness of the data is treated by the model as if it was part of the underlying structure of the phenomenon producing the data [19].

Choosing a simpler model is the same as choosing to not give outliers of the dataset the same consideration as the data that is more clustered. Although this could mean the model performs slightly worse on the validation data, the model is more robust when classifying new and unknown data because the actual patterns of the data source are learned, not noise and random fluctuations. However, a model that is too simple to capture the underlying structures of the data is called *underfitted* [19]. There is a tradeoff that needs to be done in order to strike a balance between these two modes. Data regularization is the mathematical process of choosing the simpler of two prospective models, reducing test error in exchange for possibly increased training error [7].

Model validation

For each computation performed by a model, the outcome can either be true or false, indicating whether the model did a correct classification or not. The actual classes of the data are referred to as the *ground truth*. An assessment can also determine whether a data point belongs to the intended category (positive) or not (negative). Therefore, there are four different outcomes for a given classification done by a model (see Table 2.3):

- **True Positive (TP):** The model *correctly* classifies the data point as belonging to the specific category.
- **True Negative (TN):** The model *correctly* classifies the data point as *not* belonging to the specific category.
- **False Positive (FP):** The model *incorrectly* classifies a data point as belonging to the specific category.
- **False Negative (FN):** The model *incorrectly* classifies a datapoint as *not* belonging to the specific category.

Table 2.3: The different possible outcomes when validating a classification.

	Ground truth says 1	**Ground truth says 0**
Model says 1	True positive, TP	False positive, FP
Model says 0	False negative, FN	True negative, TN

This notation makes intuitive sense for the binary case, with only two possible categories. It can also easily be generalized to classification with multiple categories. In this case, the outcomes are simply calculated for one category at a time - a positive is then regarded as the model classifying a data point as belonging to the current category.

There are three common metrics used to evaluate a classification model. These metrics all use the notation of false and true positives/negatives, which were just described. To generalize for more categories, the metrics are calculated for one category at a time.

- **Precision:** How many positives were correctly classified?

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

- **Recall:** How many correct classification were done on the actual positives?

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

- **Accuracy:** How many items were correctly classified?

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.5}$$

Cross-validation

A common practice when training models is to use N-fold cross validation. While the test set remains separate from the others, the training and validation are chosen from the same superset. In N-fold cross validation, the data set is split into N subsets, commonly five or 10, of which all but one are used for training, while the remaining one is used for validation. For each new iteration, a new subset is used for validation, with training ending when each subset has been used for validation. By evaluating the model on different validation data throughout the training process, N-fold cross-validation helps prevent overfitting [7].

2.3.3 Machine learning models

Throughout this thesis, several ML models are utilized. In this section, the specific models that have been employed will be discussed.

Kmeans clustering

Kmeans is an unsupervised ML method used for finding clusters within data. Given an input of an mxn matrix, consisting of m entries with n features each, the algorithm finds a set amount of clusters by defining k centroids (mean position of the data points in the cluster), assigning each entry to the closest centroid (using its euclidean distance), and then updating the centroid location to the mean value of the datapoints in the cluster [16]. The algorithm will output a kxn matrix with the centroids' values in each feature for all k clusters, as well as a size m array with the final cluster designation for every entry.

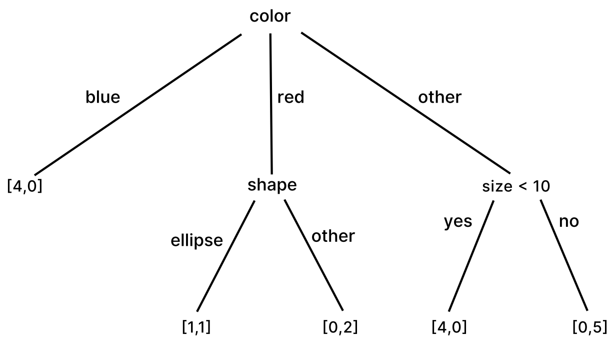

Figure 2.8: A simple decision tree for two classes, evaluated based on three features, color, size, and shape [19]. Each node evaluates a specific feature, and each branch represents an option. The leaves show the number of elements of each class in the dataset that have taken the same path.

Decision trees

In ML, a decision tree is a model in which each node represents a split based on a specific feature (see Figure 2.8). As data is evaluated, it takes one of the branches at each node, continuing this branching process until reaching a leaf [19]. These leaves denote the final decisions or classifications derived from the input features. The depth of the tree, indicating its number of splits, allows it to capture intricate details. However, there is a trade-off: while a coarse tree (with few splits) might overlook critical patterns (underfitting), a fine tree (with many splits) can become overly detailed, catching noise and reducing its ability to generalize to new data (overfitting).

Bagging

Ensemble classifiers use the combined output of many weaker classifiers (such as decision trees) to generate a stronger prediction. Bootstrap aggregating, commonly referred to as "bagging", trains a number of different trees on smaller subsets of data and aggregates the result [19]. Bootstrapping means that the subsets are created by sampling the original dataset with replacement, meaning that the same data points can be represented in several subsets. Aggregation means that the results of all the decision trees are evaluated together, for example via averaging, voting or weighted voting.

Neural networks

Artificial neural networks (ANN), also referred to as neural networks (NN), are common implementations of ML. Inspired by how neurons are thought to interact in the brains of humans and other animals, a network consists of nodes (or neurons) divided into connected and ordered layers, with edges connecting nodes in different layers. The general architecture of a neural network consists of a layer of one or more input nodes, one or more hidden layers each containing at least one hidden node, and lastly an output layer with one or more nodes (see Figure 2.9). Information from the input nodes are then passed along through the hidden layers and its nodes, propagating forward and being handled by every layer, until an output finally is computed [19].

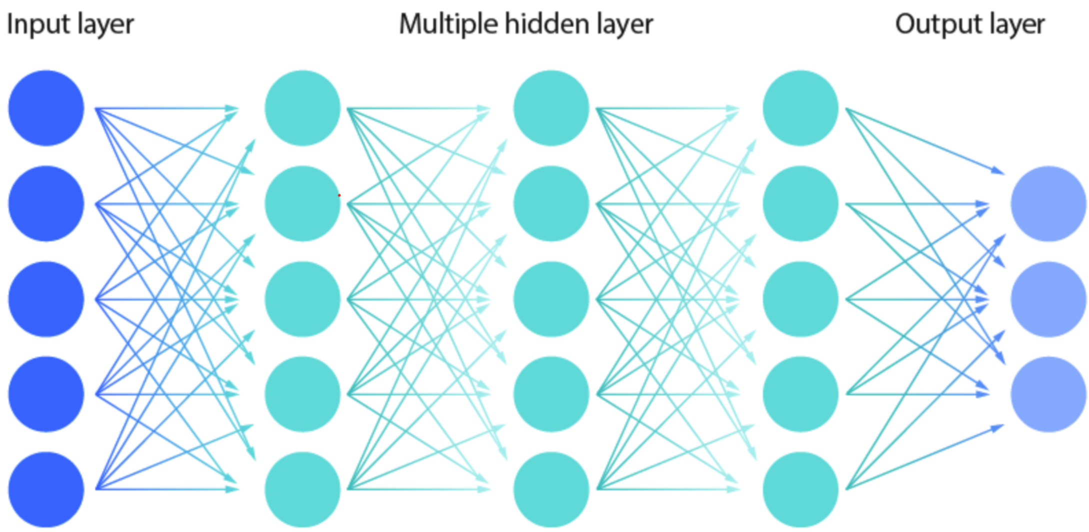

Figure 2.9: Layers of a ML neural network model [9]. Each layer consists of a certain amount of nodes, which are connected to nodes of subsequent through directed edges.

The input layer consists of one or more nodes. Each input node represents a feature in an entry. Depending on the type of network and what type of data is analyzed, the amount of nodes and what they represent can be different. For example, an input node could be a statistical feature of a time series, or it could be one R-, G- or B- value of an image pixel [19].

The nodes within the hidden layers of a network are at the core of what a neural network does. The output of a network is dependent on a web of hidden nodes that quickly becomes complex due to the density of links, and the training of a network consists of updating the parameters of the hidden nodes. Stopping the training process and looking at the parameters of nodes will seldom yield any meaningful result, due to the aforementioned complexity, but it is still important to know the main components of a hidden node, which are its weights, biases and activation function [7].

The incoming edges to a node can come either from the input layer or the previous hidden layer. Along with each incoming edge is a weight coefficient that corresponds to how important the incoming value is for the output of the node. After applying the weight coefficients to their corresponding input variables, they are summed and an offset variable, the bias, is added. After the weighted sum of the node's inputs has been computed, the result is passed as input to the node's activation function. The result from this function call is the output of the node. The choice of activation functions depends on the task, but in order to be able to model any non-linear dependencies between variables, a non-linear activation function is necessary. Some common non-linear activation functions include the ReLU function and the logistic curve function. Different functions provide different sets of advantages [7].

The output layer can be said to include two layers, one that computes the possible outputs and one that selects the output. In a classification task, the computing layer can consist of one node that computes the probability of each category. This is called *softmax*. The values from the computing layer are then passed to the selecting node, which selects the most likely category, based on the softmax output. This selection is the final output of the model [2].

2.3.4 Neural network architectures

Several types of neural networks are used in this thesis. Some common types of neural networks are multi-layer perceptrons (MLP), convolutional neural networks (CNN), and recurrent neural networks (RNN).

Multi-layer perceptron

The multi-layer perceptron (MLP) is a basic type of neural network. It is a feedforward network, meaning that the only connections between nodes are in between adjacent layers, with data flowing only "forward", only to proceeding layers. It is also a densely connected network, meaning that every single node in one layer has a connection to every single node in its adjacent layers [7].

Long short-term memory

A recurrent neural networks (RNN) is like an MLP but with the addition of feedback connections within layers, so that the output of the node (and the network in whole) depend not only on input from other nodes, but previous outputs of its own as well. Unlike the MLP, the output of a node (and in extension the output of the network) can take into account the history of the network, storing in its memory a sequence of inputs. This allows for better performance in tasks where the analyzed variables are temporally dependent, such as speech recognition and predictive text, since this allows for sequence analysis without the entire sequence being passed as input simultaneously [7].

A regular RNN can suffer from long-term dependency problems. If the output of a model is dependent on old information, there is a weight and a gradient associated with it, crucial components in the training of a network (see Section 2.3.5). To calculate the contribution of a weight, the gradient must propagate from the final output, through the layers of the network all the way to the relevant node (see Section 2.3.5). In recurrent neural networks, the weights become exponentially smaller the older the information is. This leads to the *vanishing gradient problem*, where the gradients are vanishingly small, potentially making the network unable to train itself [7]. The Long Short-Term Memory (LSTM) is an extension of the RNN intended to tackle the issues with long-term dependencies.

An LSTM has the ability to decide what information is relevant in its output. In each node, an LSTM unit is added, which in its state stores previous information. These units are called cells and consist of three gates that the stored information is passed through. The *forget* gate decides if some information should be forgotten and not used. The *input* gate decides if some new information should be added to the input before computation of the output. Lastly, the *output* gate decides what information should be added to the computed output [2].

Convolutional neural networks

Convolutional neural networks (CNN) are neural networks where at least one layer performs a *kernel convolution*. A kernel convolution is an operation where the image or time series is convolved with a kernel, and the output for a data point (pixel or time series value) is a product of a transformation of it and its neighbors. A kernel is a 1D vector or 2D matrix that is "swept" over the data and convolved, making each output data point a transformation of a neighborhood of data points. Suitable for detecting local structures and patterns within data, common applications are pattern recognition within images (2D data) and time series (1D data) [2].

Convolutional neural networks are sparsely connected networks, meaning that a node is only connected to a subset of nodes in its adjacent layers. This is because a kernel usually is smaller than the input array or matrix, this means that the output of a kernel convolution in a node is only dependent on the nodes corresponding to the data points covered by the kernel. This has advantages when it comes to pattern recognition: one is translation invariance, a pattern can be detected in any part of the data, the second is that it can learn spatial hierarchies [7].

2.3.5 Training a neural network

For the network to learn the structures and patterns of the data for classification, it needs to train on labeled and formatted data. In a nutshell, training the model means going through the data and adjusting weights and biases of the nodes based on the discrepancy between the network's output and the actual values, to incrementally optimize its performance.

The training process

The training of a neural network is done in iterations. There are two different types of iterations during training. One type of iteration is called an epoch, this refers to an iteration over the entire dataset. During an epoch, there are also smaller iterations called batches, where a subset of the training data is passed through the network. In between batches, the input weights of the nodes are updated. This updating of the weights is the actual training of the network [7].

Backpropagation

Training the model partially involves forward propagation, where data moves from the input nodes through the hidden layers to the output nodes. The model's predicted outputs are then contrasted with the actual labels in order to compute the loss. This is termed forward propagation. Backpropagation is the subsequent phase where the loss information "flows backwards" through the network to calculate how the weights of each node influence the value of the loss function. Figure 2.10 shows the general concept. Since the output of every node, including the final output node, is calculated by passing the weighted sum of its inputs into its activation function, it is possible to calculate the impact of each weight on the final cost [2].

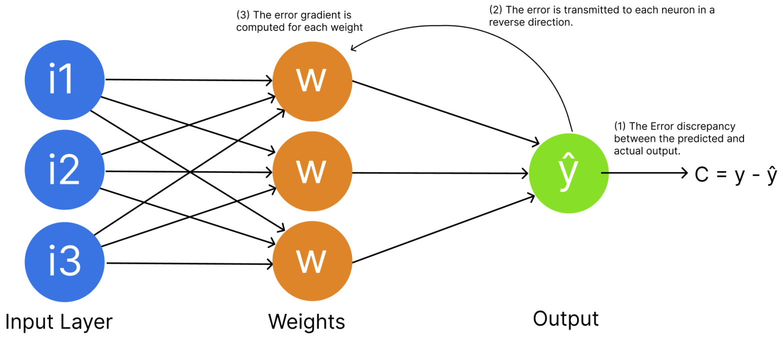

Figure 2.10: A visualization of the backpropagation algorithm.

Using Leibniz's chain rule, the partial derivative of the loss over a certain weight can be calculated. Figure 2.11 illustrates how backpropagation can be used to calculate one weights impact, by showing each node in a network as a weighted summation of inputs (z) which is passed as input to the activation function (a). The influence of $z_{m-1,1}$ is calculated by using the chain rule, obtaining Equation 2.6 [7].

$$\frac{\partial C}{\partial z_{m-1}} = \frac{\partial C}{\partial a_m} \cdot \frac{\partial a_m}{\partial z_m} \cdot \frac{\partial z_m}{\partial a_{m-1}} \cdot \frac{\partial a_{m-1}}{\partial z_{m-1}} \tag{2.6}$$

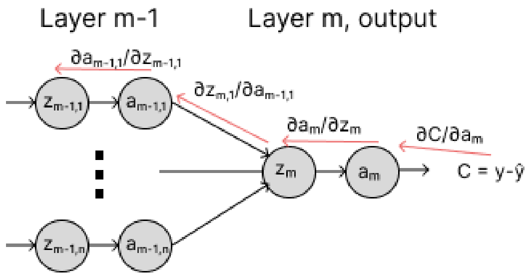

Figure 2.11: How Leibniz's chain rule is applied to neural networks, by dividing a node into a weighted summation part and an activation function part.

Stochastic gradient descent

While backpropagation computes the individual contributions of the nodes' weights to the loss, optimizers are used to compute how the weights should be adjusted. Although the relation between a weight and the loss is a function that can be traced by a two dimensional graph, it is not possible to analytically find the value of a weight so that the loss is minimized. Optimizers are algorithms that aim to calculate the weight value that minimizes this function [7]. One such algorithm is Stochastic Gradient Descent, SGD.

The gradient of the loss function over a specific weight shows in which direction the function is growing the fastest. Since you want to minimize the loss, the gradient descent algorithm simply takes a step in the opposite direction of the gradient [3]. The step size, or learning rate, defines how big this step should be. Too big of a step size and the algorithm might be unable to reach the true minimum, while a step size too small could mean it takes many iterations for the algorithm to compute the minimum. Stochastic gradient descent is an extension of the regular gradient descent algorithm, in that the algorithm runs only on small batches of data at a time. The gradient is computed for the small batch during each iteration, instead of the entire training set, significantly reducing the amount of computing needed to be done.

Early stopping

Early stopping is a data regularization strategy for neural networks that terminates training of the model if the validation loss appears to have converged to a good enough level [7]. If the loss on the validation set improves, the current set of parameters is copied and saved. These values are then overwritten for each improvement. If enough epochs have gone without sufficient improvements to the validation loss, the training is stopped and the last saved set of parameters are returned.

2.3.6 Processing of time-series data

Handling time-series data involves several key steps to make sure the information is useful for a ML model. In this section, some important steps in preprocessing of time-series data are explained: data regularization, choosing the right features, adjusting data scales, and breaking data into chunks (data windowing). These steps help in creating a model that learns well from the data without strictly memorizing it, or missing important patterns.

Data normalization

Because input variables can come from different sources and be measured in different metrics, the scales can differ from variable to variable. The scales of the motor data variables used throughout this thesis differ by orders of tens and hundreds. To easily compare and visualize the relationships between these variables, adjusting the variables to the same scales is advantageous. A common way of adjusting the scale is min-max normalization (see equation 2.7), which sets every value on a scale of 0 to 1 [21].

$$X_{norm} = \frac{X - min(X)}{max(X) - min(X)} \tag{2.7}$$

Data windowing

Windowing, to divide an arbitrary length time series data into sequences of a set length, can simplify the classification of the data. Different events may have different time spans, with start and end points potentially being difficult to extract from the data. Windowing instead changes the classification task into just classifying the new and shorter data sequences, removing the variability of duration.

The windows need to be big enough to be able to discern if an event is happening. If only part of an event is captured, the model might not recognize it. Similarly, if an event falls in two different windows, the event might not be detected in either. Having windows overlap by a certain margin, meaning that part of the current window is also covered by the subsequent window, reduces the risk of events being missed like this, as illustrated in figure 2.12 where the windows overlap by 50%. For all data points to be covered by at least two windows, a minimum overlap of 50% is required.

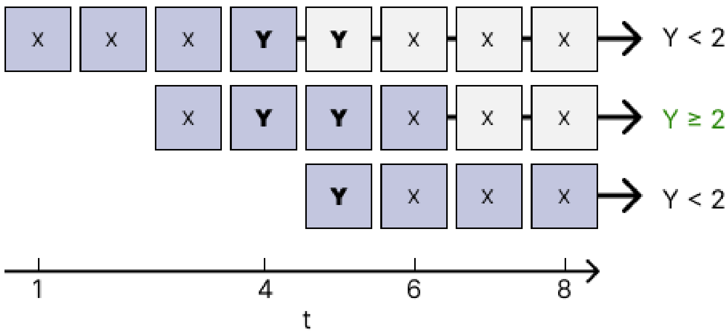

Figure 2.12: Diagram showing a sampled data sequence of events of type X and Y. The data sequence is split into windows, shown as blue boxes, of four samples' length, with an overlap of two samples, or 50%, between adjacent windows. If each window is evaluated for two samples of the Y event, only the second (middle) window properly fits the criterion.

The choice of overlap impacts how big the dataset is. Since the overlap percentage affects how many windows a given sample appears in, the same event will appear in more windows the larger the overlap is. This can affect how well the model recognizes an event, since it trains itself on many different phases of the event. While this can increase the accuracy of the model, it can also make it prone to overfitting due to excessive training on certain events.

Chapter 3

Product analysis

This chapter delves into a detailed analysis of the UX aspect of the current DMS system, particularly focusing on its leash component. Through interviews and surveys, insights are garnered from both first-time and experienced users. Topics span from the necessity of the DMS function, the convenience and safety it offers, to users' perceptions of its aesthetic appeal.

3.1 Intention of study

The main objective of evaluating the product's features was to determine whether there were adequate reasons to proceed with investigating the technical possibility of a tether-free DMS. Addressing the questions defined in section 2.2.6, the product analysis concerns the most significant usability issues of a product that is interacted with every time you use the jet surfboard. The study should therefore aim to investigate the aspects that validate the current feature, that create frustration, or that impact the enjoyment of the ride experience as a whole.

This evaluation took into account the safety, convenience, functionality, and aesthetics of the current DMS solution. It is important to note that none of these aspects, when considered individually, can fully justify either maintaining the current method of handling drop-offs, or implementing fundamental changes to it. Thus, a comprehensive assessment of multiple factors is necessary to make an informed decision. By considering all relevant aspects, the best course of action for the product's overall design can be determined.

3.2 Study with first-time users

One part of the study involved gathering user sentiments regarding the current DMS feature from people using the board for the first time. Seven individuals with varied sports backgrounds participated in this study. The participants were asked to try the board for a short while and were then interviewed.

3.2.1 Participants

Seven test users, two women and five men who had never tried jet surfing before, were invited to participate. These participants, some being friends and some recruited from the kite surfing group of one researcher, came from diverse sports backgrounds – some had extensive kite surfing backgrounds, while some had no prior experience in any surfing-related sports. The participants ranged from ages 27 to 38, with five of seven participants between 27 and 31 years old.

3.2.2 Product testing

The test consisted of two parts, an initial test ride for 20-30 minutes followed by a short interview. The procedure is shown in Table 3.1.

Test equipment and environment

The user tests and interviews took place at Vombsjön, situated to the east of Lund, on two occasions. Both days had clear skies, minimal wind, and water temperatures around 18 degrees Celsius, which were ideal conditions for jet surfing. During the test rides, the participants were equipped with a Radinn X-Sport surfboard, a battery, a wireless hand controller, a board leash, a helmet, and a life jacket. The interviews were recorded using a Samsung Galaxy A52s

Table 3.1: The procedure for user tests with first-time users.

Stage	Activity	Details
1.	Ride instructions	Inform participant about the risks to avoid and how to ride the board.
2.	Equipment	Provide participant with helmet and lifejacket, leash and controller.
3.	Test ride	The test participant rides the board for upwards of 30 minutes, alongside one of the researchers.
4.	Informed consent	Inform participant about their anonymous participation, their right to withdraw from the study, and sign consent form.
5.	Interview	Conduct and record the interview with the participant.

phone and transcribed.

Test ride

All tests were conducted individually with each participant. They were equipped with a life jacket and a helmet, and optionally a wetsuit. A board leash was handed to them, and they were asked to attach it to their preferred leg. Information about how the board leash and the hand controller works was given to the participants, along with advice on how to mount the board and get started.

Participants were instructed to avoid riding close to shore, to be gentle on the throttle and to pay attention to the battery indicator on the hand controller and return to shore if the battery was low. The participants were not given any order on how they were supposed to ride the board, they were encouraged to ride the board however they liked. However, one researcher was present to give pointers, for example on how to best get to a standing position. The ride could last for upwards of 30 minutes, but the participants were free to stop at any time.

Interview

Interviews were conducted with the participants immediately following their ride. Before starting, a brief introduction was given to explain the purpose of the study. It was emphasized that the interview was not about evaluating how well the participants could use the board, but that the main focus was on understanding their thoughts and experiences with the board and the leash. They were also informed about their rights as study participants; that their participation is anonymous, that they can withdraw at any time, how to contact the researchers, and that the audio recordings of the interviews would be deleted after the study has concluded.

The interviews followed a semi-structured approach, with 24 questions that were intended to evaluate the leash of the DMS, based on safety, convenience, functionality and aesthetics. The full template for the interview (in Swedish) can be seen in Appendix A. Since the interviews were semi-structured, the questions could be worded differently between interviews, but the same key areas were always involved.

The initial questions aimed to get an understanding of the participants previous experience with activities similar to jet surfing, such as wave surfing, kite surfing, snowboarding, or any balance-oriented sport. This was followed by asking how the ride went, if they had any issues with getting to a standing position while riding and if they had trouble with falling off. The intent was to see if there were any similarities in the sentiments of people with relevant previous experience, or those who could ride the board with relative ease.

The rest of the interview mostly intended to gather the participants' thoughts and feelings about the leash. They were asked about how the leash affected their comfort while riding, if it got in the way at any stage and if it was comfortable to wear. Rider comfort is important, these questions were intended to understand if the comfort would be significantly improved by a tether-free DMS solution.

The next questions assessed the impact of the leash on the user's perceived safety. Since the physical connection to the board via the leash is required to be able to use the board, it was thought that the physical leash could provide an increased sense of safety in itself. The participants were asked if they felt that the leash itself had a positive impact on the sense of safety for themselves and their surroundings. These questions also served to gauge user sentiments regarding the importance of the DMS function.

The participants' views on the physical leash were investigated specifically. They were asked if it was an advantage to be able to tell from the physical connection when the board was usable or not. Furthermore, their opinions on a hypothetical tether-free alternative, providing the same exact functionality as the current DMS solution, were gathered. They were asked if they thought the leash had any aesthetic impact on the board or on them, the surfer. Lastly, they were questioned regarding how the leash impacted their ride, and encouraged to share any thoughts or ideas that they wanted to express that had not already been covered.

3.2.3 Affinity diagram

After transcribing the interviews and reviewing the results to become familiar with the material, an affinity diagram was created. Important statements, feelings, and thoughts from all the interviews were written on post-it notes. Minimal changes were made to the notes, with the aim to retain the essence of the sentences while keeping them succinct. These post-it notes were then randomly placed on a whiteboard. Following the method described in section 2.2.5, similar notes were grouped together and discussed. This process went through several rounds until a satisfying grouping and labeling was done. The final affinity diagram can be seen in Figure 3.1.

Figure 3.1: Thoughts and ideas from the interviews organized in an affinity diagram. The notes are color coded based on category, with the uppermost note being the label of the category.

Resulting groups

The following final categories emerged after narrowing down and grouping the statements together.

1. **Importance of the DMS feature:** Several interviewees accentuated the importance of the DMS feature. They mentioned the safety aspect of being unable to accelerate the board when the DMS is not connected. One interviewee mentioned that it is good to be able to tell when you can use the board, just by seeing if the DMS is connected.

2. **Sense of safety:** Some participants mentioned feeling safe using the board with the DMS connected. One user mentioned that the magnet connection itself gives this sense of safety, while another said that the leash helps you know that you cannot use the throttle by accident. However, two interviewees said that the leash did not impact how safe they felt.

3. **No aesthetic impact:** The overall sentiment was that the DMS or its leash did not negatively impact the look of the board. One person said that it would be beneficial with a more brightly colored leash that is more visible in the water.

4. **Does not restrict flexibility:** When asked about the leash's impact on their freedom of movement while surfing, some interviewees said that they never had any particular issues with the leash in that regard.

5. **Affects how you interact with the board:** Some users mentioned that the presence of the physical leash of the DMS affected how they moved and interacted with the board. One participant mentioned that he was mindful of where he placed his back foot, in order to not accidentally step on the magnet of the DMS. Another one mentioned that he could only climb up from one side of the board, depending on which ankle he had attached the leash to. Furthermore, one person mentioned that the leash and the DMS was something they had to be mindful of while using the board. One person said the DMS' placement on the left side of the board was awkward because it is in the way while surfing, but that it depends on which is your back foot. A preferred choice of this interviewee would have been to place it all the way back but in the center, so as to not disfavor any surfer based on how they stand on the board.

6. **The board glides away when you fall off:** The extensive swimming that was sometimes required after falling off was seen as a safety concern and inconvenience by some of the interviewees. Two of the users with previous surfing experience expressed a wish to have a leash that is more attached to the board, so that it does not drift too far away after falling off. This leash would be in addition to the magnet on the DMS, so that the motor would still be shut down upon falling off, but the board would stay close, and be able to pull back towards you while in the water.

7. **Leash detaches:** The leash becoming dislodged and detached during rides was seen as a problem by many first time users. One user had the magnet accidentally detach twice while surfing, two others mentioned that they had the leash drag in the water and subsequently detach itself while surfing. Some users mentioned accidentally dislodging the magnet by kicking it when attempting to climb up on the board.

8. **Leash preference:** User sentiments regarding the physical leash vs. a tether-free one were divided. Many felt that the reliability of the current DMS solution was preferable. One test user stated that it is simple, safe, and works well ("it's either one or a zero"). Another said that it never fails, it will always stop the motor if you fall off. Concerns regarding a software based solution was that there is a higher risk of bugs that could break the DMS feature.

   However, other interviewees stated that the physical leash and magnet is not needed for a DMS solution, that a tether-free DMS had felt as safe as the current one. One person even said that a software solution without the physical leash would be better, another said that he would deduct points off of his review of the ride due to the leash. One user said that an LED indicator would let you know the status of the board, despite there being no leash.

9. **No issues with the leash during surfing:** Some users made it clear that they had no issues whatsoever with the leash during their test rides.

3.2.4 Takeaways from the interviews

This section will highlight the key findings from the interviews. These takeaways will serve as base for analysis and discussion surrounding the research questions.

Importance of the DMS function

The importance of the DMS function is evident from the interview responses. No tester expressed anything to the contrary. That you minimize the risk of inadvertently launching the board is important for the safety of both the surfer and their surroundings. However, the opinions on a physical vs. tether-free DMS function were divided.

Software-based vs. leash-based DMS solution

When comparing a software-based DMS solution to a physical leash, the overall preference leaned towards the latter due to concerns about potential software problems and the reliability of a physical connection. The current DMS solution is simple, it is either a "one or a zero" as one participant expressed it, and it works every time. However, not all of the participants agreed that the physical leash had to be the best DMS solution.

Some users appreciated that the physical leash in itself serves as visual feedback. Users can tell the status of the board simply from the physical connection, or lack thereof. One participant suggested that an LED to indicate the board status could be used if a tether-free feature was to be implemented. These responses suggest that the feedback aspect is important regardless of whether the DMS function has the leash or not.

Accidental dislodging of the DMS

The issues surrounding accidental dislodging of the DMS, such as kicking the magnet off when getting up on the board, or through the leash dragging in the water, suggest that the leash can be detrimental to the ride experience. The leash dragging in the water, which some users experienced, is also a safety risk, since accidentally disconnecting the DMS can cause sudden stops that throw the surfer into the water.

Inconvenience from the leash

Although most people answered that the leash did not restrict their freedom of movement on the board, some responses to other questions suggested otherwise. One tester said that his rating of the ride experience was bumped down at least two points by the leash, because of the issues it brought him during his test ride. One person mentioned that the leash is something that you have to be mindful of when riding the board, and others said that it affects how you climb up on the board, both because it is more comfortable to first kick up the leg that wears the leash, and because you have to make sure not to dislodge the magnet with your feet.

Aesthetic impression

The look of the leash was of little importance to the participants. The only opinion about the leash's look was that it could be more brightly colored, to make it easier to find when it's underwater.

3.3 Study with experienced users

The second part of the user study concerned users with previous experience of the Radinn surfboards. The 13 participants were asked to complete an electronic questionnaire that intended to capture the feelings and thoughts regarding the leash from the same aspects as the interviews: convenience, safety, aesthetics, and functionality.

3.3.1 Data collection process

Experienced board users were reached through Radinn's internal messaging platform. Data from these users was collected through electronic questionnaires. The questionnaires were designed to gather quantitative data about their experiences and their perspectives on the current DMS system. The questions were mostly close-ended with answer alternatives on the Semantic Differential Scale. However, some open-ended questions were also included to gather more detailed information beyond the SDS method, and to capture any thoughts that might not have been covered by the close-ended questions. The questionnaire can be seen in Appendix B.

3.3.2 Participants

A total of 13 respondents answered the questionnaire, of which eleven were men and two were women. The most common age group was 30-39 years old, with five respondents. Age groups 18-29 and 50-59 had three respondents each. The remaining two respondents belonged to the age group 40-49 (see figure 3.2). The questionnaire was posted on Radinn's internal Slack channel, it is therefore presumed that the responses came from individuals working at Radinn.

Figure 3.2: Age and gender distribution of the questionnaire respondents.

3.3.3 Questionnaire results

In the following section, the results of the questionnaire presented. Similar questions or themes are grouped together in the same category.

Level of experience

Participants were asked to self-assess their experience with Radinn surfboards on a scale from 1-10, with 1 being complete beginner, and 10 being expert-level. Most rated their levels of experience as an 8 out of 10. Aside from one person who rated theirs as a 4, every respondent gave a rating of 7 or higher, with two individuals giving themselves a rating of 10 (see figure 3.3). In general, individuals rated themselves quite highly, suggesting that this group has used the electric surfboard on numerous occasions or for extended periods.

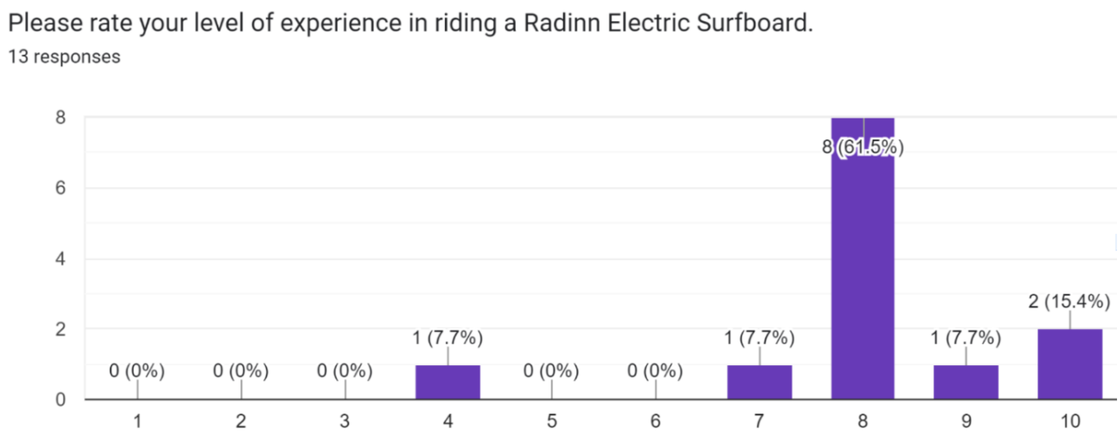

Figure 3.3: Self-assessed level of experience riding Radinn Surfboards.

Rider safety

To assess the impact of the DMS on rider safety, the respondents were asked to assess how often they experienced certain scenarios that could result in falling off the board or injury by the equipment. The answer alternatives were placed on a scale of 1-6, with a 1 meaning they have never had any problems with the scenario in question, and 6 being that it is a frequent issue.

Every respondent claimed to have experienced issues with the leash becoming entangled to some extent. Seven of the 13 participants placed a 4 or higher. Participants were asked how often they have experienced problems with the leash snapping back after being stretched as they have fallen off, which can cause personal injury due to being hit by the magnet. Responses varied greatly, there were equal numbers of ratings for 1, 2, 3, and 5. The only remaining response was a 4. One person wrote that their friend had been hit in the head by the magnet, as a result of the snap-back and gotten bruised. Another respondent also mentioned similar bruising.

The responses suggest that the main safety issue with the leash is to become entangled in it, it seems to be a relatively common occurrence for the surfers. The leash snap-back issue seemed less prevalent than leash entanglement, with most ratings between 1-3. Three individuals also claimed never to have experienced snap-back issue.

Rider comfort

The participants were asked to rate how often the leash has negatively affected their level of comfort when riding. The questions were once again on a scale of 1-6, with 1 meaning that it has never happened, and 6 meaning that it happens often.

When asked if they found the leash disruptive during surfing, the most frequent response was 3. The other responses were evenly divided between 1,2, 3, and 5, suggesting that many found the leash somewhat disruptive.

Most participants felt that the leash limited their flexibility and freedom of movement while surfing. Six out of the 13 respondents chose a 4 or a 5. Yet, most participants (seven) chose 3 or lower. Only one individual chose a 3, so there were two major groups in the response to this question.

Getting on the board

The participants were asked to rate how often they had trouble with the leash when attempting to get up on the board. The scale was once again 1-6, with 1 meaning that they have never had issues, and 6 being that it often is a problem. The answers indicated that this was a frequent issue for most users, with only one person choosing 2, and the rest choosing a rating of 3 or higher (see Figure 3.4). Two participants responded with sixes, that this is a problem almost every time they use the board. Although there were no details provided about how the leash was problematic, it nevertheless seems to cause issues specifically when climbing up on the board.

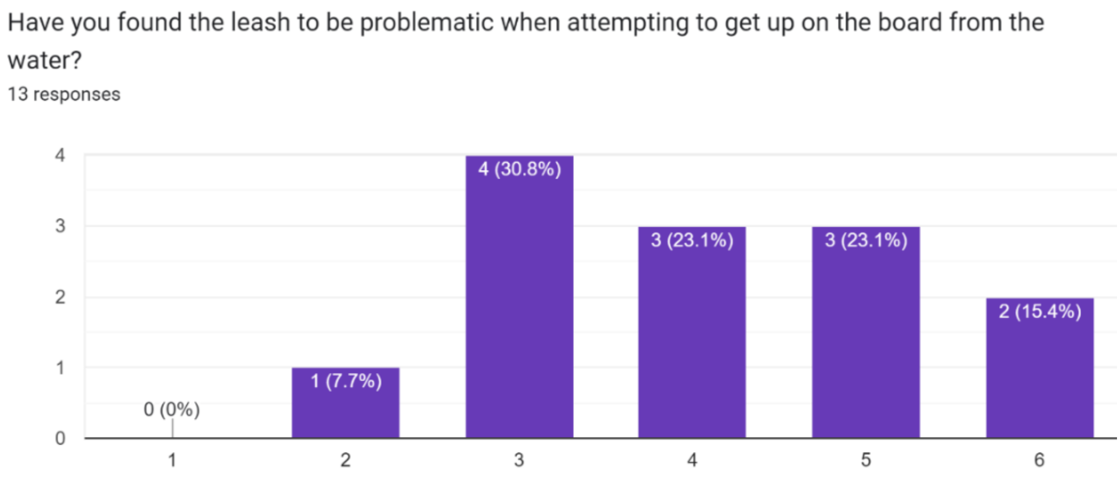

Figure 3.4: Respondents' experiences with the leash when mounting the board.

Personal harm vs. leash use

Participants had been asked to rate their desire to ride without the leash (despite potential risks) on a scale from 1-6, with 1 indicating "definitely not" and 6 meaning "absolutely". The most common choices were 1 and 6, both receiving three votes. Options 2, 3, and 5 received two votes each, while option 4 got one vote. While responses varied, the polarization between "definitely not" and "absolutely" stood out (see Figure 3.5).

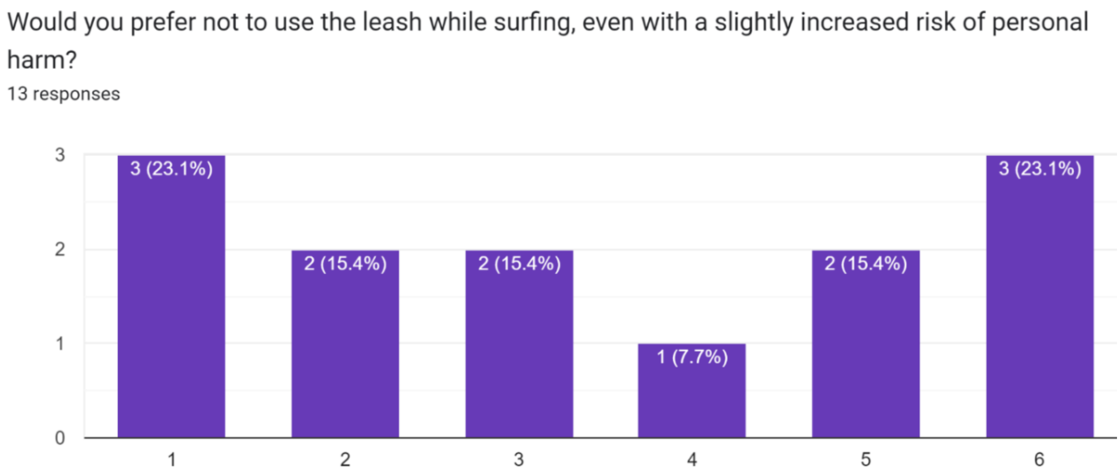

Figure 3.5: Respondents' attitudes towards riding without a leash and the subsequent increase in risk of personal harm

Aesthetic impression

Most individuals, 10 out of 13, felt neutral about the leash's aesthetics. However, two respondents had strong opinions. One felt that it significantly detracted from the overall impression, while another believed it greatly enhanced it. One other respondent thought it slightly diminished the overall impression.

Overall satisfaction

In terms of how the leash impacted satisfaction with the board, participants had 5 options ranging from greatly decreasing to greatly increasing satisfaction. Out of the 13 respondents, 10 felt neutral, no impact in either direction. However, one person answered that the leash greatly reduced their satisfaction, one answered that it somewhat decreased it, while one said it somewhat increased it.

Advantages and specific concerns regarding leash

About the advantages of being physically connected to the board, one person felt that the connection prevented the board from drifting too far when they fell at high speeds. Another respondent commented on the board's weight, suggesting that being attached to the board might not be advisable. They also emphasized the importance of a DMS function to ensure safety when the leash is detached.

3.3.4 Takeaways from questionnaire

The perspectives of experienced jet surfers are valuable in the assessment of the leash. More experience riding the board means more times dealing with the leash, giving slightly more weight to their assessment of its issues and advantages. In this section the takeaways from the questionnaire will be presented.

Importance of the DMS function

Many of the respondents highlighted the need for a DMS for this product. However, some said they could see themselves riding without a leash, despite an increased risk of personal injury. The people who most wanted to ride without the leash (who answered a 5 or a 6) generally rated highly on questions regarding frequency of issues with the leash, with some exceptions.

Others had the opposite idea about riding without a leash. Among those who least wanted to ride without the leash (who answered 1 or 2), there was a bigger variation in the answers regarding its issues. The general tendency among those respondent was that the leash does not cause them many problems. Another respondent in this group had answers that indicated great annoyance with the leash, high rating on most issues, but still stated that they would not ride without it. This person's answers suggests that they think the DMS is still much too important to go without.

Software-based vs. leash-based DMS function

The questionnaire did not directly ask whether a software-based DMS solution would be acceptable. Nonetheless, responses from the survey might still provide an indication of how such a solution could be perceived by experienced users. Six out of the 13 participants scored 5 or higher when asked if they would consider operating without a leash, even if it would slightly increase the risk of harm, while 7 would not. This effectively divides the respondents into two groups. One participant noted that the DMS offers reassurance that the drivetrain (the mechanical components propelling the board forward) is deactivated when not attached, which indicates that the visual feedback provided by the physical leash might still be a crucial component.

Safety issues

Accidental dislodging of the DMS is a problem even for more experienced users. Two people mentioned that the magnet often pops off during riding, and that the positioning of the DMS can cause it to be kicked off by your backfoot, or dragged off by the water. Dislodging the DMS will stop the board rapidly, which can pose a risk to the surfer and their surroundings.

Both entanglement and snap-back issues come with varying degrees of severity. Some participants specifically noted that the magnet caused injuries like bruises and cuts when they fell. Leash entanglement could also heighten the risk of accidentally detaching the magnet from the DMS, causing sudden stops.

Inconvenience with the leash

The majority of people, eight out of 13, said that the leash is problematic when attempting to get up from the water (rated 4 or higher, out of 6). Getting up on the board was the situation where people had the most issues, with an average rating of 4. Six people said that it quite often (rated 4 or higher) restricts freedom of movement when surfing, and five people said it quite often restricts freedom of movement in the water. That the leash causes inconvenience can also be gathered from the fact that some people want to ride without the leash. Had it not been uncomfortable, the leash most likely would not have mattered.

Aesthetic impression

The vast majority of the participants were neutral to the look of the leash, and believed it neither detracted or added to the aesthetic impression of the board. However, there were two non-neutral answers, that both believed it detracted from the board's look, both strongly and less so. It seems like the issues with the DMS mainly involve it being in the way, causing issues at various points during the ride.

3.4 Summary of user study

This section presents the aggregation of the takeaways from both the interviews with the first-time users and the survey conducted with the more experienced users. While a part of the group found value in the presence of the leash, primarily for its perceived safety benefits, many of the participants expressed a more negative opinion on it. A common sentiment revolved around the disruption of the surfing experience due to the leash.

3.4.1 The need for a DMS function

Across both user groups the need for a DMS function was clearly stated. Concerns with the current solution revolve around the physical leash.

3.4.2 Software-based vs. leash-based DMS function

The opinions regarding the idea of having a tether-free DMS solution instead of the physical leash were divided. The results from both the survey and the interviews show that there is basically an even distribution between people who prefer either solution, with many opinions landing in between. In the group of more experienced users, a few stated that they could definitely consider riding without a leash, even if it came with a higher potential risk of personal harm, while others had the complete opposite opinion. Some inexperienced users liked the simplicity and reliability of the leash and expressed no need for a tether-free alternative, while others said a tether-free alternative would be preferable or just as good.

Advantages of the physical leash, as gathered from the interviews, is the simplicity and reliability of it. It is a simple matter of "one or zero", the DMS is either connected or not, as simple as possible. The status of the DMS can also be read from the magnet connection, a simple visual indication of the board's status.

3.4.3 Rider convenience

The leash can be inconvenient when surfing. For those who expressed inconvenience with the leash, the most frequent issue was when attempting to mount the board. That this issue was expressed by first-time users, as well as more experienced users, suggest that it is a problem for most people.

Accidental dislodging

Both new and experienced users mentioned problems with the DMS accidentally dislodging, causing the board to come to a stop. First-time users that were interviewed mentioned kicking the magnet when they attempted to get up on the board, while others had it detached by the leash dragging in the water. This same sentiment was expressed by some experienced users in free text answers to the survey, but this was a small minority, only two out of the 13 survey participants.

Entanglements and snap-backs

Every questionnaire respondent claimed to have had issues with the leash becoming entangled in some way, either in the board or one of their body parts. The issue did not seem as prevalent among the first-time users that were interviewed, although this could be because of the short duration of their ride. The same can be said for snap-backs, which a majority of the questionnaire respondents claimed to have experienced at some time, but no interviewees had.

3.4.4 Aesthetic impact

There were only two participants that believed that the leash detracted at all from the appeal of the board. The only other opinion about the look of the leash was that it should be more brightly colored, for increased visibility. The overall opinion seems to be more neutral.

Chapter 4

Machine learning modelling

This chapter will describe the process of constructing the machine learning models aimed at accurately identifying rider presence, along with the outcomes of these models. The first section will outline some of the inherent requirements of the task that must be addressed by the constructed model. In section 4.2, the data under analysis will be detailed, including its frequency, scale, and source. The two strategies for the tasks are explained in section 4.3, along with the specific challenges that affect each strategy. Section 4.4 details how some of the data needed for the on-board detection approach was gathered. The process commences with an exploratory data analysis, explained in section 4.5, where an initial examination of the data is conducted to identify potential patterns or features that can be utilized in the subsequent machine learning modelling. In sections 4.6 and 4.7, the complete processes of both strategies are detailed. These sections reiterate what data is used, how the data is pre-processed and what features are extracted for usage in the machine learning modelling. The different classes are defined and presented, along with the models and the setup for the simulations. Lastly, the results for each strategy are presented in numbers and graphs, followed by what can be gathered about the strategies based on the results.

4.1 Requirements and safety concerns

As a fail-safe mechanism, the performance of a DMS is of utmost importance. If a mistake is made and the software-DMS falsely believes the surfer has fallen off or believes they are present, the surfer could be put in danger. In addition, the time a decision takes to make, including computation time, has an impact on the performance of the DMS and the surfer's safety.

4.1.1 Importance of accurate surfer detection

As a security feature, the machine learning models need to be able to correctly identify the surfer's presence at all times. The consequences of falsely presuming that the surfer has fallen off is that the board stops abruptly, running the risk of throwing the surfer off the board. Additionally, if the rider has fallen off and this has not been registered, the surfer could send the board away at high speeds, with no way of controlling it, just by accidentally pressing the throttle. Evidently, minimizing false negatives and false positives is key for the task.

4.1.2 Consequences of incorrect detection

The probabilities for the two described scenarios actually happening differ slightly. As a surfer becomes more experienced, it can be presumed that less time is spent falling off the board and more time is spent actually surfing. The surfer will spend more time running the risk of having the board mistakenly stop the motor, than accidentally pressing the throttle and sending the board away with no one on board. Therefore, when comparing two models, the first priority is to compare how well they can classify regular riding of the surfboard, with second priority being how well they classify drop offs.

4.1.3 Time sensitivity

The described solutions needs to be able to identify rider absence quickly, in order to be able to stop the board from gliding away too far. A model that is too complex would add computation time to a time sensitive task, which could impede the ability to stop the motor in time. Complex and computation heavy models might also be difficult to implement on the board's hardware. The importance of complexities and computation times of the models are secondary to the classification results, as implementing a model on board is not within the scope of this thesis.

4.2 Data insights

In order to understand the purpose of the choices made when constructing the machine learning model, it is important to understand the source of the data, along with its structure. This section will present the data that is analyzed. It will include a presentation on where the data comes from, how it was gathered and what it represents. The section will also present some inherent questions posed by the data, and state how they inform the decisions made during model construction.

4.2.1 Data source

In this thesis the focus is on three key metrics: the electrical RPM (eRPM) of the water jet motor, the current passed through the motor, and the status of the DMS. Table 4.1 shows the scales and units of the metrics. Normally, this motor data is sampled at a rate of 1 Hz, meaning a snapshot of the motor current and eRPM are taken once every second.

Table 4.1: Raw data used in the machine learning task

Metric	Scale	Unit
Electrical RPM (eRPM)	0 – 33000	eRPM
Motor current	0 – 335	Ampère (A)
DMS	[Detached, Attached]	Boolean

The data associated with the DMS is of binary nature and is not analyzed the same way as the motor data. The DMS status only indicates if it is connected and disconnected. It is used in this thesis in order to find places in the data where potential fall-offs have happened.

Fetching data logs from Radinns' database can be done in two different ways: the first approach involves the return of multiple logs, which can be filtered and aggregated in various ways, such as through input-queries, filtering, and aggregating based on date, size, and so forth, followed by downloading. Alternatively, when only a small number of logs are desired, single logs can manually be downloaded as individual files.

4.2.2 Limitations and challenges

This section delves into the limitations and challenges associated with the motor data used in the analysis and the development of the machine learning model.

Temporal resolution

As previously mentioned, the data in the pre-existing ride logs has been sampled at 1 Hz. A relatively low temporal resolution, meaning the number of discrete samples per time unit, makes the task of identifying short-spanning events, such as falling off the board or accelerating with no one on board, more difficult. Subsequently, due to the resolution, it is possible that rapid fluctuations or steep slopes are overlooked, which might otherwise be useful in pattern recognition.

With one second in between samples, and proper classification based on only single samples seeming unreasonable, any sort of pattern recognition would take at least one second. Time is of the essence for detecting when no-one is on board; if it is too slow, it is possible to inadvertently apply acceleration until the controller loses its connection to the board, with the board travelling much further away with no one on board. In Figure 4.1, the distinction between 1 Hz and 20 Hz sampling frequencies is demonstrated. By comparing the same time series snippet with both frequencies, it's evident that lower sampling rate inhibits the ability to detect shorter potential patterns and information.

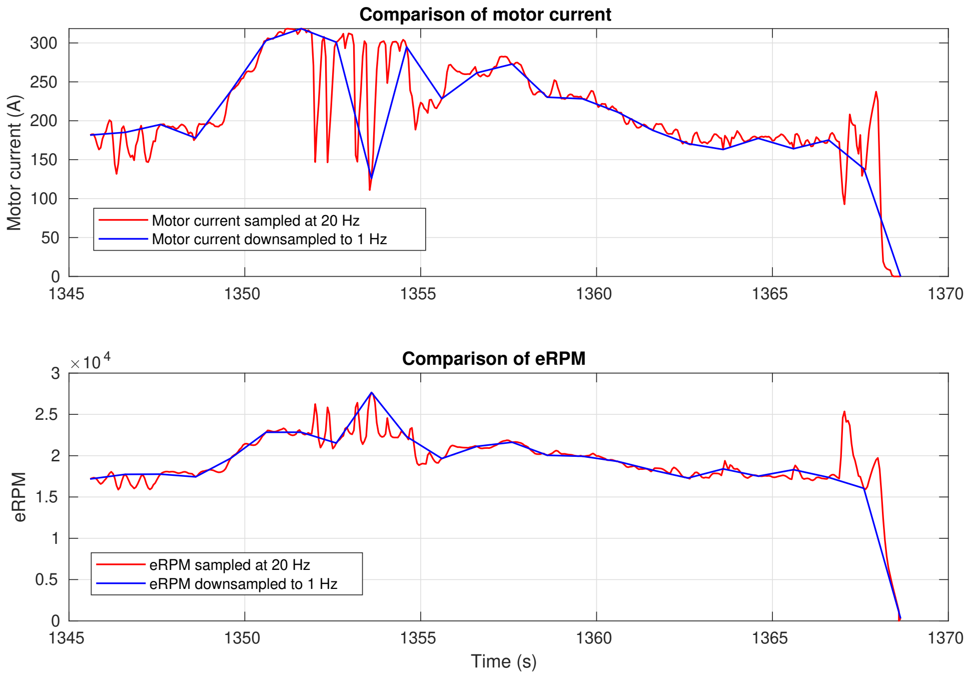

Figure 4.1: A comparison of 1 Hz and 20 Hz sampling frequencies using the same time series snippet.

Annotation and labeling

Another potential issue with the data from the logs is that it is not annotated. There is no labeling that says what is happening when the DMS is detached, so discerning between when a surfer has fallen off and when they have jumped off deliberately or disconnected the DMS manually becomes a strict matter of interpreting the data from the motor logs. Similarly, the weather conditions are unknown, meaning waves and wind might have an effect on the speed of the surfboard which is unknown for someone that just looks at the motor data.

4.3 Detection strategies

During the project, two ideas regarding drop-off detection were explored. One idea was to identify when the rider is in the process of falling off the board, by comparing the motor data from instances where the DMS was triggered against those instances where it was not. The second idea was to identify the difference in the motor data between when a rider is on the board and when they are not. These ideas are called fall-off detection and on-board detection, respectively. The following section will explain the ideas in detail, along with some potential limitations in both approaches.

4.3.1 Fall-off detection

The fall-off detection approach should identify a possible pattern in the motor data for when the surfer is falling off the board. This could be how the data looks when the surfer is losing balance, or how the data differs when the board is slowing down with or without a surfer on board. This approach can make use of the preexisting data available in the database, as the status of the DMS is stored in the logs and can be used to extract a large amount of data from moments when the surfer has potentially fallen off.

This approach is affected by all the limitations mentioned in Section 4.2.2, both the temporal resolution of the logs and the lack of annotation have an effect. Due to the lack of annotation, part of making a model that can meaningfully distinguish fall-off events means to also find a distinction between an actual fall-off and a deliberate disconnection of the DMS, strictly by looking at the motor data. The low temporal resolution presents a challenge both in discerning events in the data, but also affects the speed of the model. The time between samples directly affects the speed of the classification, as new classifications can only be made as fast as new data arrives.

4.3.2 On-board detection

The on-board detection approach is based on the hypothesis that the additional friction and drag resulting from the surfers weight impacts how much eRPM is generated at a specific level of current. As a result, the eRPM-current relationship may differ when someone is on the board compared to when not. A motor that has to propel both the board and the surfer will have to work harder than one without the surfer, resulting in a higher eRPM per ampère of current in the motor for the surfer-less board.

This approach could possibly serve as both an on-board detection mechanism and a fall-off detection mechanism. In theory, a ML-model could distinguish if the surfer is applying acceleration as they fall off and deactivate the motor. Additionally, it would account for situations where the board is accidentally launched while the surfer is in the water, or attempting to get up on the board.

A big challenge for this approach lies in collecting the data. For this task, there needs to be data collected from a surfer-less board, so that models can train on both the on-board and off-board scenario. The only natural surfer-less data is generated in the short instances where the board is accidentally launched when the surfer is about to mount the board after connecting the DMS, or when accelerating when falling off. Since the DMS status cannot be used as an indicator of off-board or on-board scenarios, this data needs to be gathered manually.

4.4 Collection of high frequency data

The ride logs from all boards are normally sampled at a frequency of 1 Hz, making the preexisting data only suitable for the low-frequency aspects of this thesis. In addition, the on-board detection task needed data from a surfer-less board, which was not available in the database. Therefore, this data needed to be generated and collected manually.

High frequency data was collected on three occasions, once in Malmö Harbor, once in

Vombsjön and once in Lomma and Habo Ljung. Batteries, equipment and two X-Sport Radinn boards were provided by Radinn. In order to allow for collection of data at higher sampling frequencies, custom firmware was created and installed on the boards. The new firmware had a sample frequency of 20 Hz compared to the previous 1 Hz.

To collect data from a surfer-less board, one board was ridden by a researcher while simultaneously controlling a second board with an additional controller. The board without a rider had the DMS connected and the leash strapped to the board to hold it in place. To collect *on-board* data, a surf board with the custom firmware was ridden normally. When collecting data it was crucial to ensure clarity in the data obtained from each ride. During the *on-board* data collection, efforts were made to avoid falling off. When categorizing rides for training machine learning models, it is vital that the categorization is as accurate as possible. To generate a diverse range of data while both on-board and off-board, the rider maneuvered the boards at varying speeds and executed numerous accelerations. When a ride was finished, a phone was connected and the ride log was uploaded to the cloud.

4.5 Exploratory Data Analysis

To begin understanding the data and how the tasks could be performed, an exploratory data analysis (EDA) was conducted. The primary objective of the EDA was to uncover patterns, structures, and outliers within the dataset. The discovered traits within the data could then serve as a guide for the classification tasks.

4.5.1 Analyzing Current vs. eRPM Relationship

To understand how the eRPM performs across different motor current values, data from multiple rides stored in the cloud was collected and analyzed. The eRPM values were sorted into bins, according to the corresponding current value at the same sampling instance. In order to illustrate the distribution of eRPM values for each possible motor current value, the mean, median, and 25th and 75th percentiles were calculated. Figure 4.2 shows the graphs for these values, with the x-axis showing the current and the left y-axis showing the eRPM. The orange bar graph in the same figure shows the number of samples gathered at every current value.

Since the eRPM is roughly proportional to the acceleration of the board, and the motor current level is proportional to how hard the user presses on the throttle, this graph can be said to show how much speed can be expected at any given level of throttle. A narrower eRPM distribution therefore means that a safer estimate of speed can be made based on the current level, while a wider distribution means a greater uncertainty. The graph shows that at higher levels of motor current the relation between current and eRPM is more linear, with little variation. The eRPM distribution widens at lower current levels, with a difference of between 15 and 20 thousand between the 25th and 75th percentiles at currents 35 A and lower.

The wider distribution of eRPM values at lower currents can be attributed to many things. One is that many surfers test the equipment on dry land before heading into the water, revving up the motor to see that everything works properly. These "dry runs" result in a comparatively high eRPM gained for the throttle that is being applied by the user, since

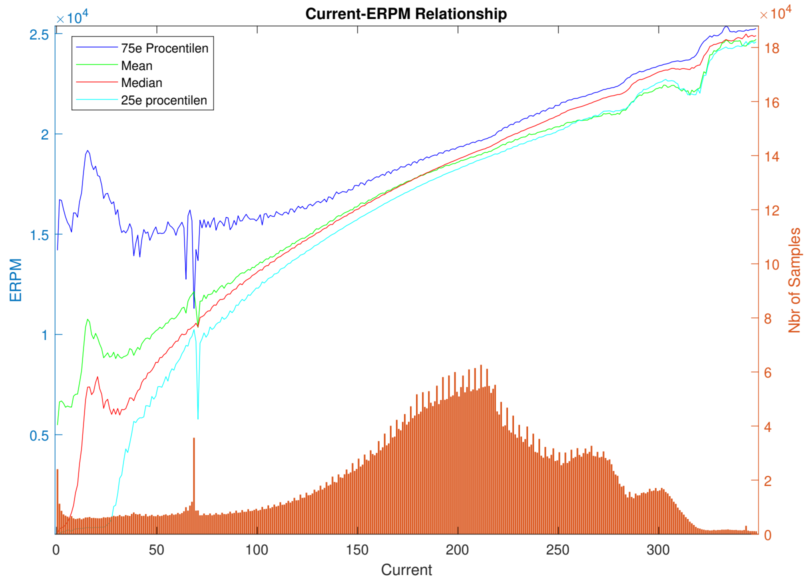

Figure 4.2: Visualization of the distribution of eRPM for each possible level of current. This plot illustrates the relationship between Current and eRPM, showcasing various performance metrics, such as the 75th percentile, Mean, Median, and 25th percentile values. The orange bar diagram, plotted against the right axis, shows how many samples there were for each level of current.

there is no resistance added by the flow of water through the impeller, only the mechanical and electrical resistance within the board. The same principle as above applies when the jet intake takes in air during rides, which can occur in bursts when the water is turbulent. Additionally, the lower current levels are mostly spent in deceleration or acceleration. When the surfer releases the throttle, the current to the motor is stopped almost instantly, but the board will glide to a stop, maintaining a flow of water through the intake, which will rotate the impeller, maintaining eRPM even with no electrical current to the motor.

Some characteristics of the data can also be learned from the bar graph which shows the amount of samples at each current level. Higher current values, from 100 A and upwards, are more represented in the graph than values below. While lower values, with some exceptions, are sampled 10000 times or less in total, all values between 150 A and 300 A are sampled over 20 000 times, rising to upwards of 60 000 around 200 - 220 A. Considering that most time spent riding the board is at a steady pace at moderate to high speeds, it makes sense that there are more samples at higher current values than lower ones. The lower current values mostly occur when either accelerating from or decelerating to a complete stop, which most surfers spend little time doing, especially experienced surfers.

There is a sharp peak in number of samples at approximately 70 A. When surfers want to get back to shore, they can use a ride mode which sets the board to a constant low speed. In this mode, the current sits at 70 A, with eRPM sitting around 10 000. This explains both the increased number of samples at this current level, but also the sharp dip in all eRPM metrics except the median.

4.5.2 Rider detection techniques

The exploratory data analysis also served as a way of investigating the feasibility of the fall-off and on-board detection tasks, as well as trying to find similarities and differences in the data that could aid in the tasks.

Fall-off detection

The DMS data from each ride log was used in order to find instances where the DMS had been disconnected. Each ride log was split into windows consisting of three samples, with an overlap of two samples between adjacent windows. If a DMS disconnection was detected within a time window, this window was placed in a separate category. Due to the large amount of idle time in the logs, with motor current and eRPM values below a threshold (circa 0.08 after normalization), windows that had no significant activity (two or more eRPM or current values below the threshold) were placed in their own category as well. This meant that all the ride logs had been windowed and placed into one of three categories of windows: *idle*, *riding*, or *DMS triggered*.

Figure 4.3 shows a scatter plot of the (normalized) minimum and maximum values of motor current and eRPM. Each dot in the graph represents a window, with the color of a dot corresponding to the category of that window. It can be seen that when the minimum values or the maximum values are plotted against each other, the data points follow a curve, looking similar to Figure 4.2. Looking at the "DMS triggered" windows in the lower left plot, it can be seen that there is a bigger variance in the values, compared to riding windows. The DMS windows also have lower minimum eRPM and current levels.

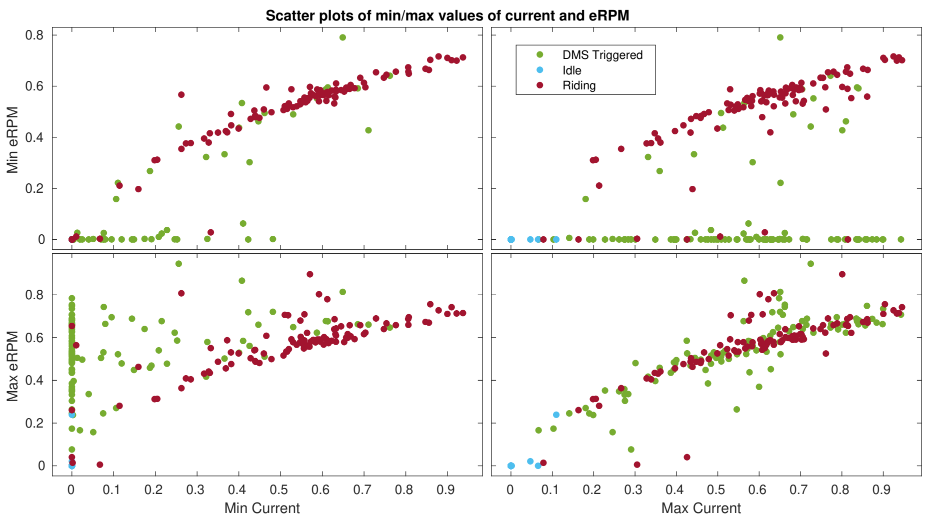

Figure 4.3: Scatter plots of the minimum and maximum motor current and eRPM values.

The trend of a value, meaning the last sampled value subtracted by the first value, over the window span can also show differences between different types of windows. A simple analysis of the trend of a value could be that a positive or negative value means acceleration or deceleration, while a value of zero means speed has remained constant. A scatter plot between the trends of eRPM and motor current is shown in Figure 4.4. By observing the diagonals, the histograms show that the riding and idle windows are generally centered around zero in both eRPM and motor current. The windows with DMS triggers generally show more of a negative trend, which makes sense because the DMS disconnecting means that the motor is shut off and an obvious deceleration to zero.

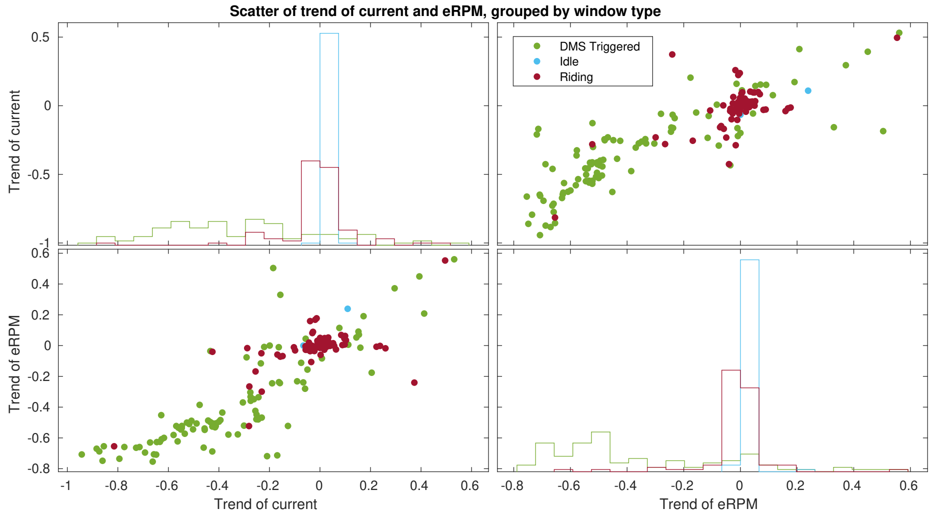

Figure 4.4: Scatter plots of the trends of eRPM and motor current for idle windows, windows where the DMS has triggered, and windows of regular riding of the board. The plots in the negative diagonal show the distribution of each metric.

On-board detection

Investigations were conducted to determine whether it was possible to identify differences solely through the analysis of eRPM and motor current data, by comparing multiple statistical parameters from unmanned board as well as a board with a rider. The parameters considered included the average, median, and 25th and 75th percentiles of the data, which were collected and processed using the same method as described in subsection 4.5.1. Data from unmanned and manned runs were plotted alongside each other for comparison, as shown in Figure 4.5.

Examining Figure 4.5, there is an observable difference when comparing the data with a rider on the board to the data without a rider. While the distinction may not be very pronounced, variations can be observed at different parts of the motor current spectrum, but most part of the spectrum is covered when combining all four statistical values. These initial results are promising for the potential development of an ML model to distinguish whether someone is on the board or not. These four parameters alone may not be sufficient,

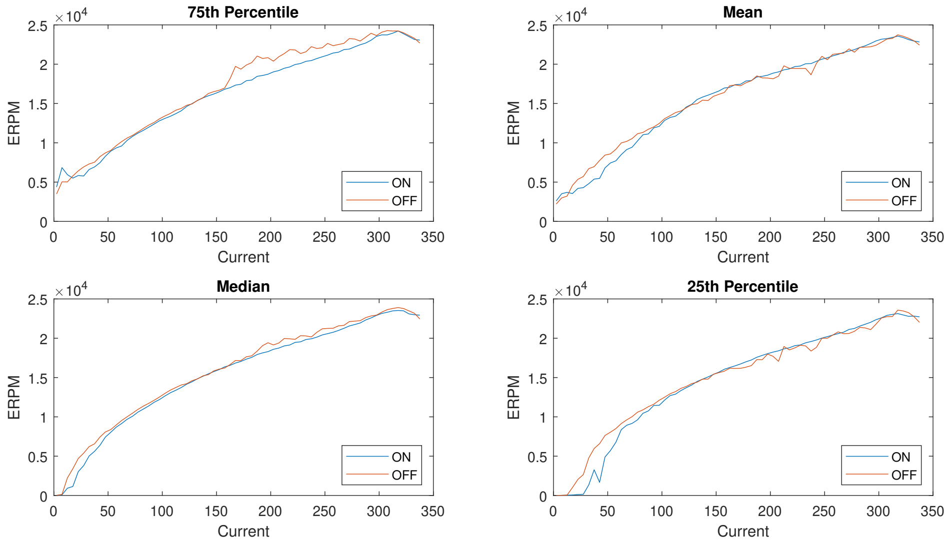

Figure 4.5: Visualization eRPM values for each current value with and without rider: The four plot illustrates the relationship between Current and eRPM, each subplot show various performance metrics, such as the 75th percentile, Mean, Median, and 25th percentile values.

and additional features, as discussed earlier, should be added.

Examining the difference between the eRPM and motor current values also lends some credibility to the on-board detection idea. Figure 4.6 shows a grid of scatter plots of the 20th percentile, 80th percentile and mean value of the difference for windows of 10 samples. The distributions shown on the negative diagonal (upper left to lower right) of the grid show that the distributions differ slightly. The on-board windows generally have a wider distribution that also has a lower mode than the off-board distributions. The windows chosen are a small randomized subset of on- and off-board data, showing that off-board riding tends to have a higher ratio of eRPM:current, but with a noticeable overlap between the two groups.

4.6 On-board detection

In this section, a comprehensive assessment of how to create and investigate a method for on-board detection, as described in subsection 4.3.2, is presented. This method was evaluated by navigating through the data processing stages and testing various machine learning models, ultimately yielding results that demonstrate the model's potential performance on real ride logs.

4.6.1 Data collection

All data for this approach had to be collected manually (see Section 4.4), because no data from a surfboard without a rider existed beforehand. Two 45-minute runs without a surfer present were conducted, generating logs sampled at a higher frequency. In addition to the rides

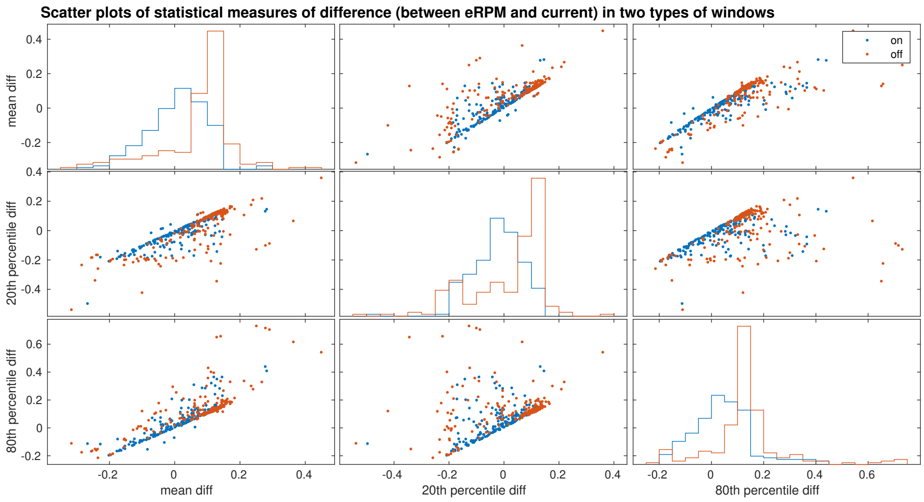

Figure 4.6: Scatter plot of mean, 20th percentile and 80th percentile of the difference between eRPM and current for windows of size 10. Grouped by category of window, on-board or off-board. The negative diagonals show the distributions for each feature.

categorized as 'no surfer present,' six 30-45 minute rides were performed and categorized as 'surfer present' Both types of rides generated high-frequency logs.

4.6.2 Pre-processing

Since the ride logs used for this task were generated manually, no filtering of logs was necessary. To be utilized as training and validation data, the logs had to undergo normalization according to Equation 2.7. The normalized data was then divided into windows. A window span of 10 samples, equating about 450 milliseconds, and 20 samples, equating about 950 ms, was chosen to represent two suitable timeframes for data analysis, ensuring that each window possessed a sufficient number of data points. For both window sizes, datasets were created with a 50% and 75% overlap. This corresponded to a 5 sample and a 8 sample overlap for a window size of 10 samples, as well as a 10 and a 15 sample overlap for a window size of 20 samples. The overlap of 50% is the smallest overlap percentage that ensures an even coverage of all values in the time series, while a 75% overlap is of interest due to the possibility of a faster acting DMS when new windows appear at a higher rate. As seen in Table 4.2 there were four windowing configurations, each being chosen with the objective of achieving a balanced approach, ensuring that the sliding window covered a relevant timeframe while also preserving computational efficiency and data integrity.

4.6.3 Features

Features were extracted from windows of sequential data, each encompassing two distinct variables: eRPM and motor current values. Each data window was initially divided into these two categories for separate analysis (Eq. 4.1), and a difference metric was calculated to represent the difference between the eRPM and current values (Eq. 4.2). Subsequently, 27

Table 4.2: The sliding window configurations that were applied to the high frequency data from the database.

Configuration	Config. 1	Config. 2	Config. 3	Config. 4
Window span (ms)	450	450	950	950
Window size (samples)	10	10	20	20
Overlap percentage	50%	80%	50%	80%
Overlap (samples)	5	8	10	16

features were extracted from the windows, nine features each for the current, eRPM and difference metrics (Eq. 4.3). The first and last raw values from each window were also initially incorporated as input to the ML model (Eq. 4.4). This resulted in 32 inputs to the ML model (Eq. 4.5). The derived features from the windowed data include basic statistical measures like median, percentiles, mean, minimum, and maximum, in addition to the secant (trend), calculated for both the original values and their differences across the time window. By extracting the features from windows, each window forms a new feature window representing a new entry in the dataset.

$$W_i = \{(eRPM_{i1}, C_{i1}), \ldots, (eRPM_{iL}, C_{iL})\} \qquad \text{Data window extraction} \tag{4.1}$$

$$D_i = \{eRPM_{i1} - C_{i1}, \ldots, eRPM_{iL} - C_{iL}\} \qquad \text{Difference calculation} \tag{4.2}$$

$$F(W_i) = \{\text{Median}(W_i), \text{Mean}(W_i), \ldots, \text{Secant}(D_i)\} \qquad \text{Feature extraction} \tag{4.3}$$

$$F'(W_i) = \{eRPM_{i1}, C_{i1}, \ldots, C_{iL}, D_{iL}\} \qquad \text{First and last diff. values} \tag{4.4}$$

$$\text{Input}_i = \{F(W_i), F'(W_i)\} \qquad \text{Forming ML Model Input} \tag{4.5}$$

- W_i: Represents each data window, containing L eRPM and current values.
- D_i: The difference metric, indicating the difference between eRPM and current values within window i.
- $F(W_i)$: Denotes the extracted features from each window, including statistical and secant measures.
- $F'(W_i)$: Represents the first and last value from each window..
- Input_i: Forms the final input to the ML model by combining extracted features and the first and last values from the windows.

To reduce complexity and streamline the performance of the machine learning models, the selection of features from the initial set of 32 inputs was necessary because not all features added anything useful, they did not improve the models' classification abilities. Hence the number of features were narrowed down the from initial 32 inputs to 21 features, seven features each for the motor current, eRPM and difference. This selection was achieved by utilizing the mRMR algorithm, enabling the identification of the most pertinent features for the task while eliminating those with limited value. The final features are shown in Table 4.3.

Table 4.3: The final features used in the models.

Metric	Features
Motor current	Mean, median, 20th percentile, 80th percentile, minimum, maximum, trend
eRPM	Mean, median, 20th percentile, 80th percentile, minimum, maximum, trend
Difference	Mean, median, 20th percentile, 80th percentile, minimum, maximum, minimum derivative

4.6.4 Defining classes

There were three classes defined for this task: *idle*, *on-board*, *off-board*. All time series from the ride logs contain idle segments and periods when the board is not receiving any input from the rider. The lengths of these sections can vary, ranging from just a few seconds to a half hour or even more, and thus represent significant part of the total collected data. Idle windows are categorized by the absence of motor activity. If the motor current and eRPM are very low (lower than 0.03 after normalization), there is no way of discerning if the surfer is on-board from the data. If the data came from a ride without a surfer and was not idle, it was labeled as *off-board*. Similarly, if the data came from a ride with a surfer and was not idle data, it was labeled as *on-board*.

Table 4.4: The different window types that were defined for the on-board detection task.

Window type	Motor activity	Surfer on board
Idle	No	-
On-board	Yes	Yes
Off-board	Yes	No

4.6.5 Models

Various models possess the ability to identify and classify different types of data, contingent upon the nature of the input data. The models listed below were used for this classification task.

1. Decision tree
2. Bagged decision trees
3. CNN
4. LSTM
5. CNN + LSTM

4.6.6 Simulation

To address the issue of potential overfitting, arising from an excessive amount of *on-board* data compared to *off-board* data, an equal number of windows from both classes were utilized during training. The quantity of *on-board* windows was approximately five times that of the other, thus, all *off-board* windows were used, and a selection of *on-board* windows was chosen for model training. However, the specific choice of *off-board* windows significantly influences the model's performance in subsequent window classifications, thereby causing variation in the model's validation results depending on the window selection. This issue will persist as long as a size difference exists between the amounts of data collected from each activity. To ensure robust results from the simulations, multiple runs were executed for each model. Consequently, each model was trained and validated 20 separate times, using a different set of 20 randomized data windows for each run, across various overlaps and window sizes.

4.6.7 Results

In this section, the performances of the ML models are presented. The metrics, revealing the statistical and numerical efficacy of each model, are followed by a presentation of real-world results, indicating instances and scenarios where they perform better and worse.

Performance metrics

The metrics used to evaluate the performances of the models were the precision and recall of *on-board* windows and *off-board* windows. The bar diagrams in Figure 4.7 show the performances of all the models, grouped by window type and performance metric. The bars show the mean values for each metric, with the error bars showing the 95% confidence intervals. For the individual diagrams, as well as the exact numbers of each metric, see Appendix D.

There are little differences in performance scores between each window configuration, although the configurations with a higher overlap percentage perform slightly better. There is a symmetry between the recall scores and the precision scores of opposite window types. If there is a low recall score for one type of window, there is a similarly low precision score for the other type of window. Both types of windows have either higher precision and lower recall, or vice versa.

The mean precision score for the *on-board* windows and the mean recall score for the *off-windows* both range between 85,8% and 92,6%. The mean recall score for the *on-board* windows and the precision score for the *off-board* windows range between about 70,2% and 77,7%. Confidence intervals are generally slim for all models except the LSTM models, with the intervals for the decision tree and bagged tree models being nearly negligible.

It can be seen from the diagrams that the bagged decision tree model performs best according to most metrics in most window configurations. Looking at the recall metric for *on-board* windows, the bagged trees model consistently has a mean score of over 0.75, meaning it correctly classified 75% of the *on-board* windows. Similarly, it has the best recall score for *off-board* windows for two configurations (10-sample windows with 50% overlap, and 20-sample windows with 50% overlap).

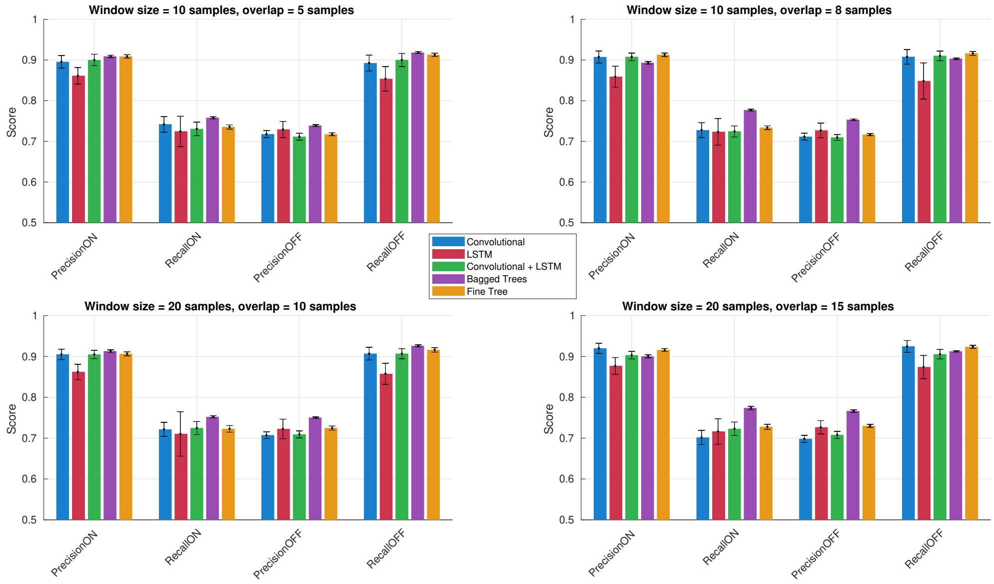

Figure 4.7: Diagrams showing the mean scores and 95% confidence intervals of recall and precision, for all window configurations, models, and different types of windows.

Visualizations of classified windows

It is important not only to consider the performance scores but also to take into account what would be categorized as *off-board* in a real-life scenario. The sequence shown in Figure 4.8 is a snippet of a ride log, where each data window has been classified by a bagged tree classifier. The ride log has been windowed into chunks of 10 samples, with an overlap of 8 samples (rounded up from 75% of 10). The blue graph shows the normalized motor current, and the orange graph shows the normalized eRPM. The surfer is riding as usual, slowing down at times without falling off. The snippet ends with the surfer falling off the board at high speed without releasing the throttle. The DMS disconnection is marked in the graph by a vertical black bar.

The red areas mark the windows that the model has classified as *off-board*. The figure shows some misclassifications, since only the falling off at the end can be said to be potential actual *off-board* data. The misclassified windows are mostly windows where the board is accelerating or decelerating strongly. A look at another part of this ride can be seen in Figure 4.9, where similar misclassifications can be seen.

It is important to see how well the model can recognize both classes of windows. Figure 4.10 shows a snippet of a ride log of a board without a surfer. The red areas show the ticks that are covered by windows that the bagged tree model has misclassified as *on-board*, mistaking an empty board for a board with a surfer on it.

4.6.8 Takeaways from on-board detection

Reading the bar diagrams in Figure 4.7, it can be assumed that the model is skewed towards classifying windows as *off-board*. In practice, if an *on-board* window is misclassified, it is going

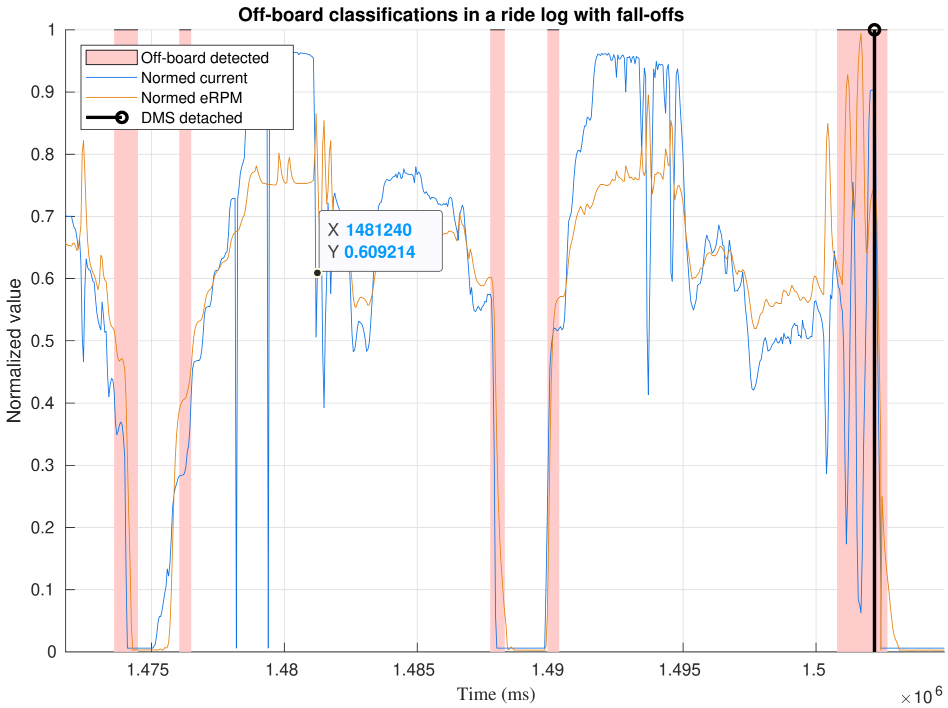

Figure 4.8: Section of a ride log where the red areas mark the time stamps covered by windows that the classifier has identified as *off-board*. The vertical black bar marks where the DMS has been disconnected.

to be misclassified as *off-board* Since *idle* windows are defined by having no significant motor activity, the overlap between them and the other window types is low. If the model is skewed towards *off-board* classifications, it will correctly classify the actual *off-board* windows many times (high recall of *off-board* windows), but also mistake *on-board* windows as *off-boards* (low precision for *off-board* windows). Conversely, the remaining *on-board* windows will be mostly correctly classified (high precision for *on-board* windows) but still have low recall, due to the aforementioned misclassifications.

The *off-board* classifications seems to be skewed towards heavy acceleration and deceleration. Hence, even *on-board* accelerations and decelerations are misclassified as *off-board* data. Figures 4.8 and 4.9 show that there usually are steep declines or inclines, or fluctuations in the eRPM and current values wherever the model has found *off-board* windows.

4.7 Fall-off detection

The following section will describe the work done on the fall-off detection task. The process of data collection, pre-processing, feature extraction, class definition and model selection is detailed. Numerical results as well as an illustration of real-world performance are presented. The section ends with a summary of the takeaways from the results of this task.

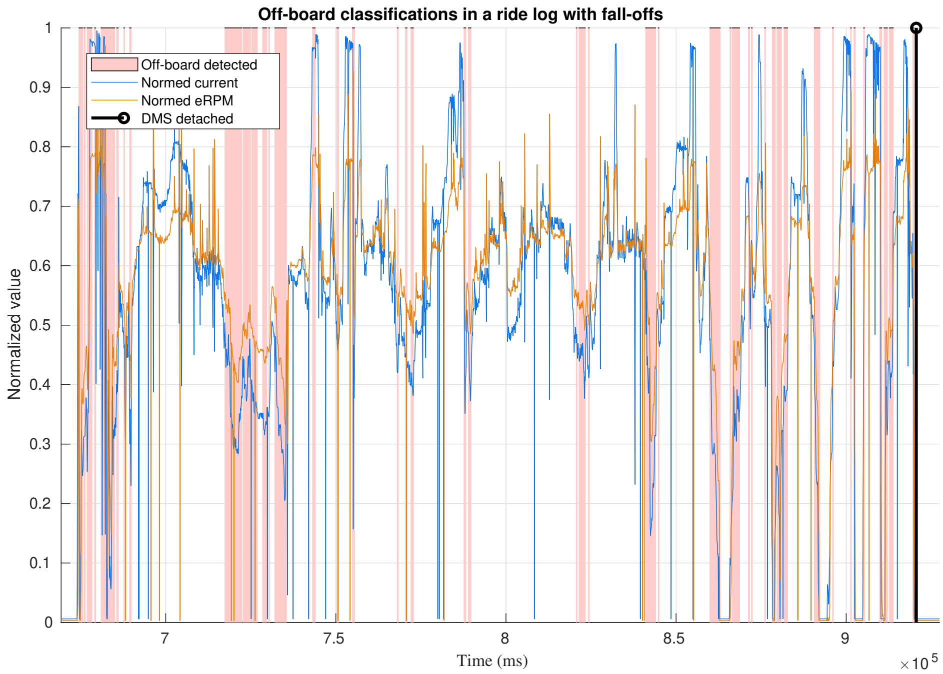

Figure 4.9: Another view of the same ride as Figure 4.8. Red areas mark where the classifier has found an off-board window.

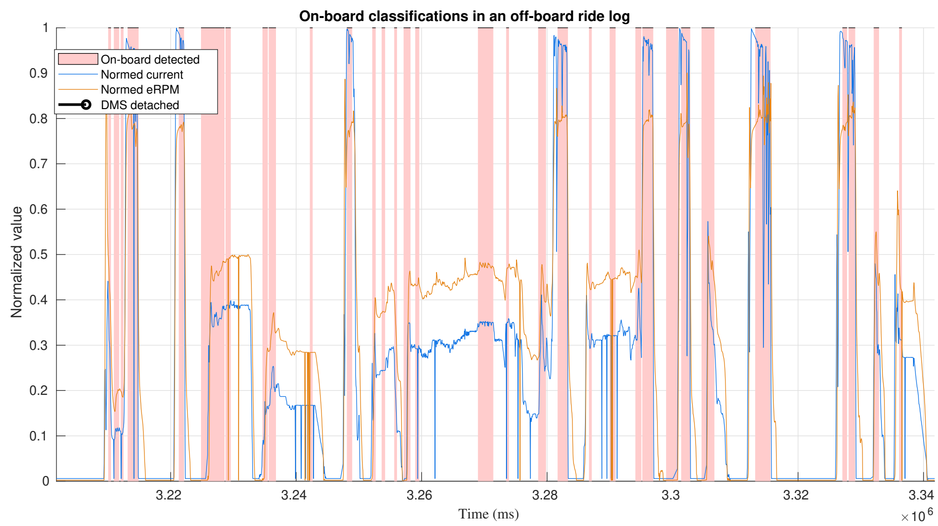

Figure 4.10: Snippet of a ride log *without* a surfer on-board. The red areas mark where the classifier has found an *on-board* window, meaning that it mistakenly thinks that a surfer is riding on the board.

4.7.1 Data collection

The data used for this task was all from pre-existing ride logs stored in the Radinn analytics database. This data was collected by querying the Radinn database for ride logs in which all the eRPM, current and DMS data was available. This resulted in an initial set of 12 000 ride logs.

4.7.2 Data pre-processing

Due to the large volume of data, some filtering had to be done to ensure, to the best possible ability, that low quality logs were filtered out. After filtering, roughly 6600 out of the 12 000 initial logs remained. The remaining logs were then min-max normalized and windowed. Four different window configurations were tested.

Filtering unusable data logs

Due to little insight into the quality of the ride logs beforehand, as well as the large volume of logs, it was necessary that low quality logs were filtered out from the final dataset. Luckily – due to the large amount of ride logs that was gathered from the database – corrupted or faulty logs could simply be discarded without too much of an impact on the size of the dataset. The logs could therefore be filtered out based on quite strict criteria to determine if they were usable.

The intention of the filtering was to not have logs that could be from testing on dry land, had corrupted data, or data that was otherwise not representative of situations where the surfer can fall off. Ride logs were refused if the DMS was activated or deactivated fewer than five times, if useful motor data (higher than a minimum threshold) was gathered at fewer than fifty samplings, or if the maximum current and eRPM of a ride were too low. These logs were deemed too likely to be of a test ride or containing too little valuable data to make it worth the risk. If the samples had somehow been scrambled into non-chronological order, or the eRPM samples and current value samples were out of sync, it was deemed easier to discard a log than to manually fix it. Lastly, the logs where the towing mode, as described in subsection 4.5.1, had been used for more than 50 seconds were also discarded, for the same reason. As a result, an initial set of 12 000 ride logs was reduced to roughly 6600 usable logs.

Normalization

All ride logs were min-max normalized according to Equation 2.7.

Windowing

The normalized ride logs were windowed. The windows were defined by number of samples, not a time span, with the overlap being calculated as a percentage of the window size in samples. Table 4.5 shows the four different window configurations that were applied to the low frequency data from the database. The shortest window span of 1000 ms is the smallest possible that actually covers a range of time, while the maximum window span of 3000 ms was chosen because it was thought that this was the absolute longest span in which a single fall-off event could be captured cleanly, without also capturing a significant amount of data from other events.

Table 4.5: The sliding window configurations that were applied to the low frequency data from the database.

Configuration	Config. 1	Config. 2	Config. 3	Config. 4
Window span (ms)	1000	2000	3000	3000
Window size (samples)	2	3	4	4
Overlap percentage	50%	66,7%	50%	75%
Overlap (samples)	1	2	2	3

4.7.3 Features

Due to the small amount of data contained in each window, raw values, along with the calculated difference (the normalized eRPM subtracted by the motor current) were used as input for the machine learning models, instead of features (see Table 4.6).

Table 4.6: The values that were used as input for the fall-off detection models for windows of size L.

Metric	Features
Motor current	$[Current_1, ..., Current_L]$
eRPM	$[eRPM_1, ..., eRPM_L]$
Difference	$[eRPM_1 - Current_1, ..., eRPM_L - Current_L]$

4.7.4 Defining classes

Defining classes for the training and testing set was needed in order to be able to assess the performance of the models. In order to find fall-offs as well as possible, there needed to be four predefined classes. These classes were: *idle*, *riding*, *fall-off*, and *disembark*.

Window types

A window was defined as *idle* if there was no significant activity in the motor. Significant activity in motor was defined in the code as the normalized eRPM and current being below a normalized threshold of 0.05, which equates to about 17 A for the current and 1500 eRPM. This value was chosen after observation of the data, noting what the levels were when the throttle was being applied. The odd samples that are below the threshold are those taken exactly as the board has started accelerating, or when it is coming to a stop. The significance of defining *idle* windows is that these are times when there is no motor data to analyze, hence there being no way of distinguishing between someone being on the board or off the board.

A window was defined as a *DMS window* if the DMS was disconnected close to the end of the window, within 500 ms of the final timestamp. This was done in order to possibly find patterns in the data *before fall-offs*. However, as a large portion of the DMS disconnections are intentional, there needs to be an attempt at distinguishing between intentional disconnections and fall-offs, hence there being two types of DMS windows.

If a window was neither idle or a *DMS window*, it was defined as *riding*, meaning that there is a surfer on the board riding normally. The *riding* window is mainly defined by what it is not, namely an *idle* window and a *DMS* window. Only one of these thresholds are defined by the motor data, a riding window cannot be idle. A *DMS window* is not defined by the motor data, but is defined by the DMS disconnection. Only the distinction between a *fall-off* and a *disembark* is defined by the motor data. As such, there is a great variety in how an arbitrary *riding* window or *DMS window* can look, leading to a higher risk of potential overlaps between these categories. See Table 4.7 for an overview of the types of windows that were defined.

Table 4.7: The different types of windows that were defined for the fall-off detection task.

Window type	Motor activity	DMS disconnected
Idle	No	-
Riding	Yes	No
Fall-off	Yes	Yes
Disembark	Yes	Yes

Distinguishing between DMS windows

As previously mentioned, a distinction had to be made between deliberate and accidental disconnections of the DMS. The results of the EDA, as seen in Figure 4.4, showed that there appeared to be two separate clusters of data windows, where one cluster of windows is trending more negatively in both eRPM and current. It seemed logical that deliberate DMS disconnections would be preceded by the board slowing down, hence a negative trend in both current and eRPM. In order to define a dividing line between the two clusters, unsupervised machine learning was used to find different trends yielded by the features.

Windows that preceded DMS disconnections were featurized, calculating 10 features each for the current, eRPM and the difference (eRPM subtracted by the current). The features were: Median, 20th percentile, 80th percentile, Mean, Minimum value, Maximum value, Variance, Minimum derivative, Maximum derivative, Trend (last value of the window subtracted by the first value of the window). The feature windows were passed as input to the kmeans algorithm to create two clusters of datapoints. This was done for windows of two samples and three samples.

The results of the kmeans clustering, using the windows of three samples can be seen in Figures 4.11 - 4.13. The graphs show the centroids of the two clusters for every feature derived from the current, eRPM and difference. The difference features do not indicate any significant clustering, but it can be seen for both the current features and the eRPM features that the two clusters differs a lot for the 20th percentile, minimum value, minimum derivative and trend features.

The resulting cluster centroids for windows of size two can be seen in Figures 4.14-4.16. Similar to the clustering of three-sample windows, the main distinguishing features between the clusters are the trends. The cluster centroids do not differ markedly for the features of the difference value, but appear so only due to the small scale of the y-axis.

Because of the small size of the windows, there is likely a significant correlation between the features where the biggest differences between the two clusters are found, especially for

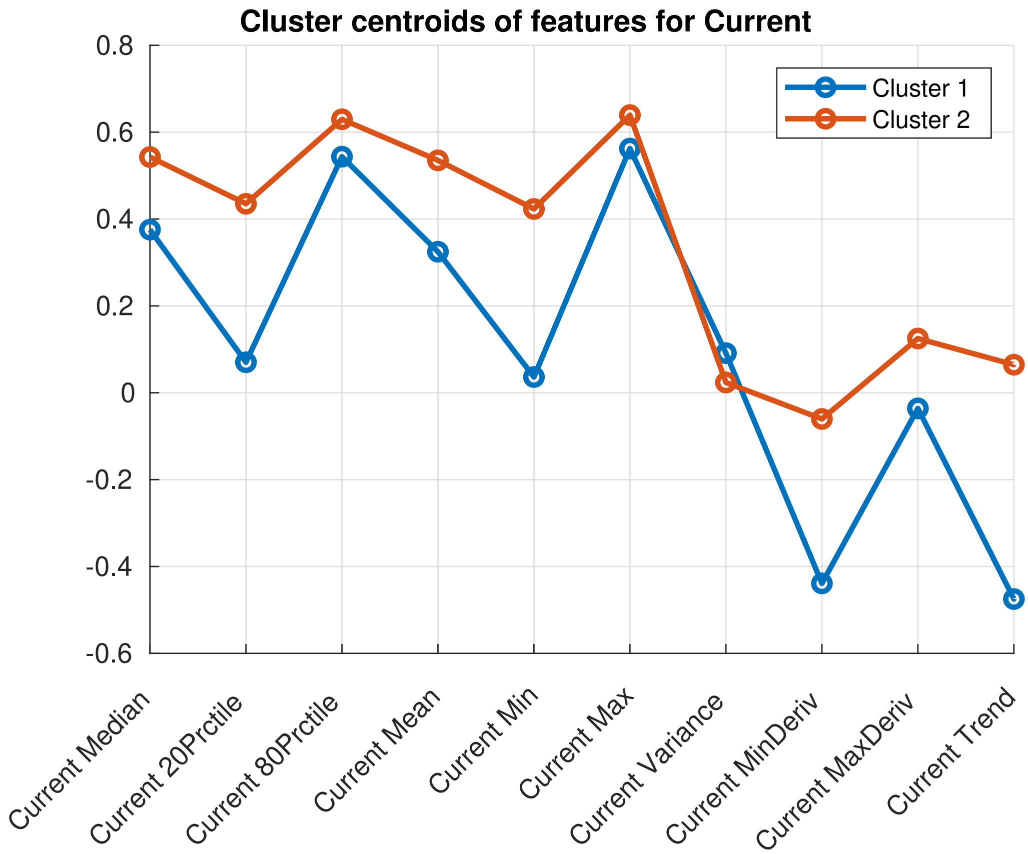

Figure 4.11: Centroids of the two clusters for windows of size THREE, in the features of the motor current data, found by the kmeans algorithm.

the windows of size two. With a time of one second in between samples, a significant deceleration made on the board is likely covered in large part by two samples, meaning that the trend and the minimum derivative are correlated. The 20th percentile value and the minimum value are also correlated, since there are not that many values to compute a percentile from. A strong deceleration will also lead to a low minimum value for both motor current and eRPM, meaning there is a correlation between these two features.

The output of the kmeans algorithm seems to be in line with the findings from the EDA (see Figure 4.4), meaning the kmeans algorithm has likely found the same clusters. The cluster centroids were used in order to define the thresholds between what was more likely to be a controlled disembark, and an actual fall-off. The trends and the minimum values of the eRPM and motor current were defined as distinguishing features for these clusters, with only the trend being used for windows of size two, due to the aforementioned correlations between some features.

Since the distribution of values within a cluster can differ, the threshold should be defined with respects to the characteristics of both distributions. For each defining feature, the standard deviations of both clusters in that feature were calculated. The threshold was placed midway between one standard deviation off the mean (see Figure 4.17). The threshold between (presumed) *disembarks* and *fall-offs* for the windows of size three can be seen in Table

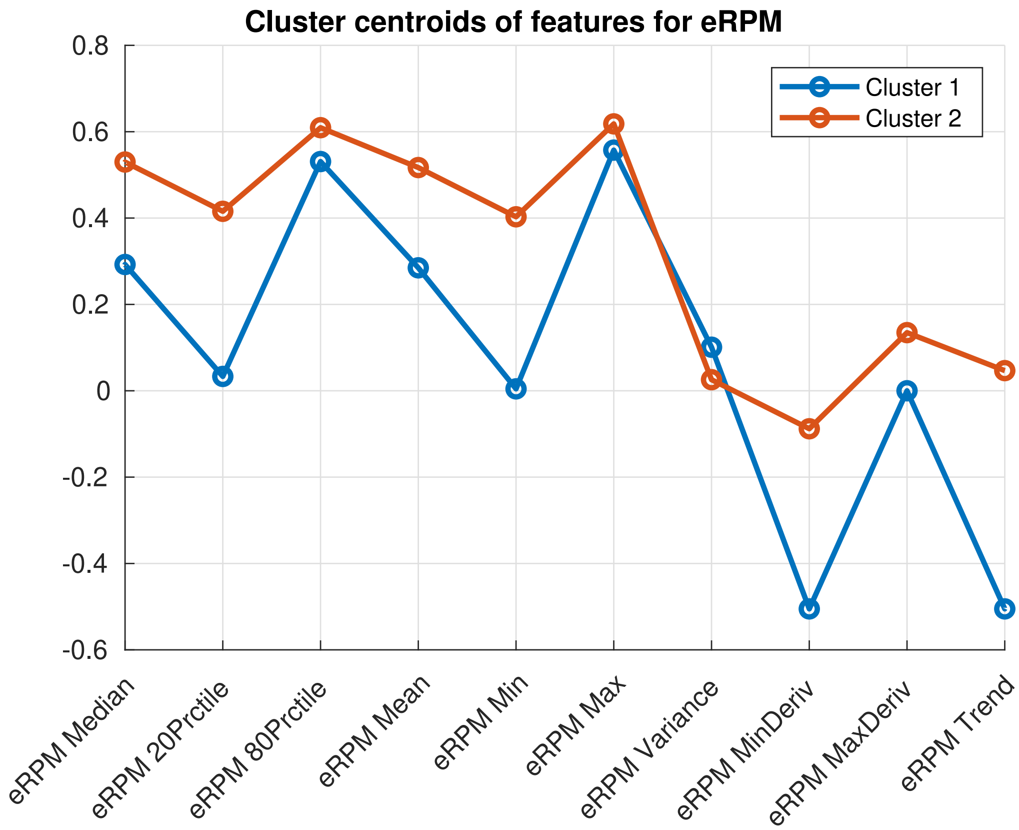

Figure 4.12: Centroids of the two clusters for windows of size THREE, in the features of the eRPM data, found by the kmeans algorithm.

4.8. The threshold for windows of size two are presented in Table 4.9.

Table 4.8: Thresholds defined for windows of three samples.

Metric	Feature	Disembark	Fall-off
Motor current	Trend	< -0.189	≥ -0.189
	Min. deriv.	< -0.250	≥ -0.250
eRPM	Trend	< -0.256	≥ -0.256
	Min. deriv.	< -0.321	≥ -0.321

Table 4.9: Thresholds defined for windows of two samples.

Metric	Feature	Disembark	Fall-off
Motor current	Trend	< -0.301	≥ -0.301
eRPM	Trend	< -0.320	≥ -0.320

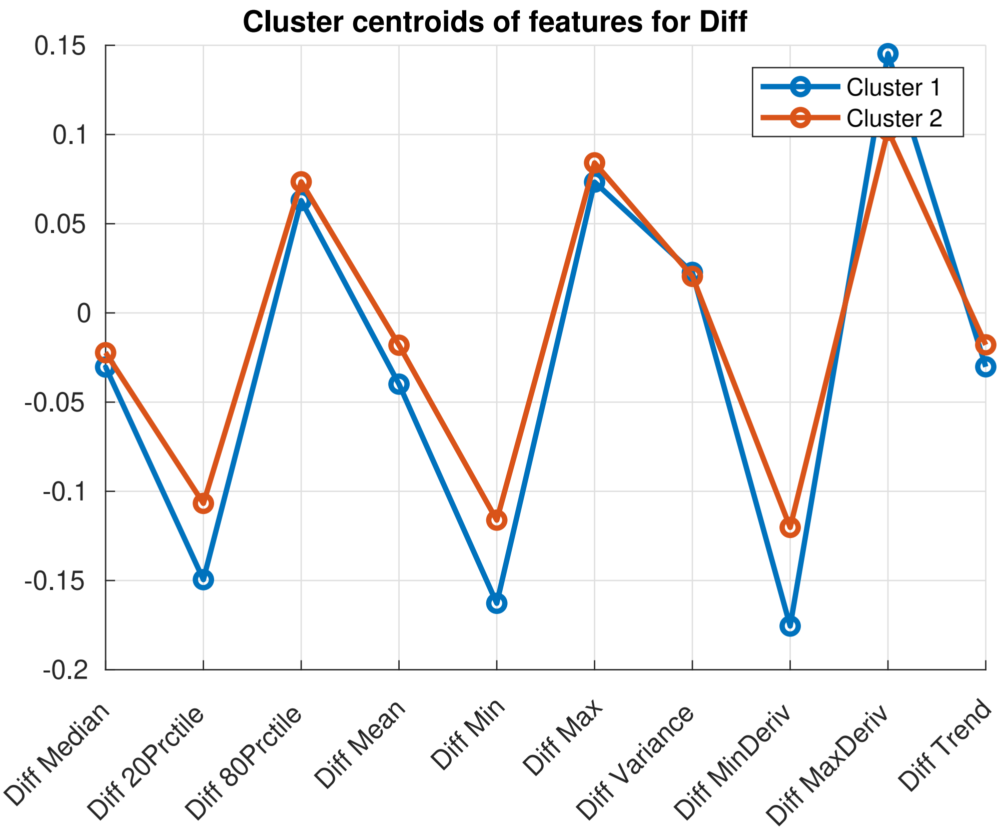

Figure 4.13: Centroids of the two clusters for windows of size THREE, in the features of the difference (eRPM - current), found by the kmeans algorithm.

4.7.5 Models

The same types of models as in the on-board detection task was used for this task, see Section 4.6.5.

4.7.6 Simulations

The fall-off detection task was performed 20 times for each model and sliding window configuration. The training set for each model was a balanced dataset consisting of an equal number of *idle*, *riding*, *fall-off*, and *disembark* windows. A new training set was created before each iteration, ensuring that results were not just the result of a lucky choice of training data. The testing data was the same for each iteration.

4.7.7 Result

The performance of the models are presented in this section. The performances are first presented as *recall* and *precision* scores for the *riding* and *fall-off* window types, showing how well these two window types can be distinguished. To visualize what type of patterns are recognized as *fall-offs* and *disembarks*, a ride log is plotted, where the classifications are marked.

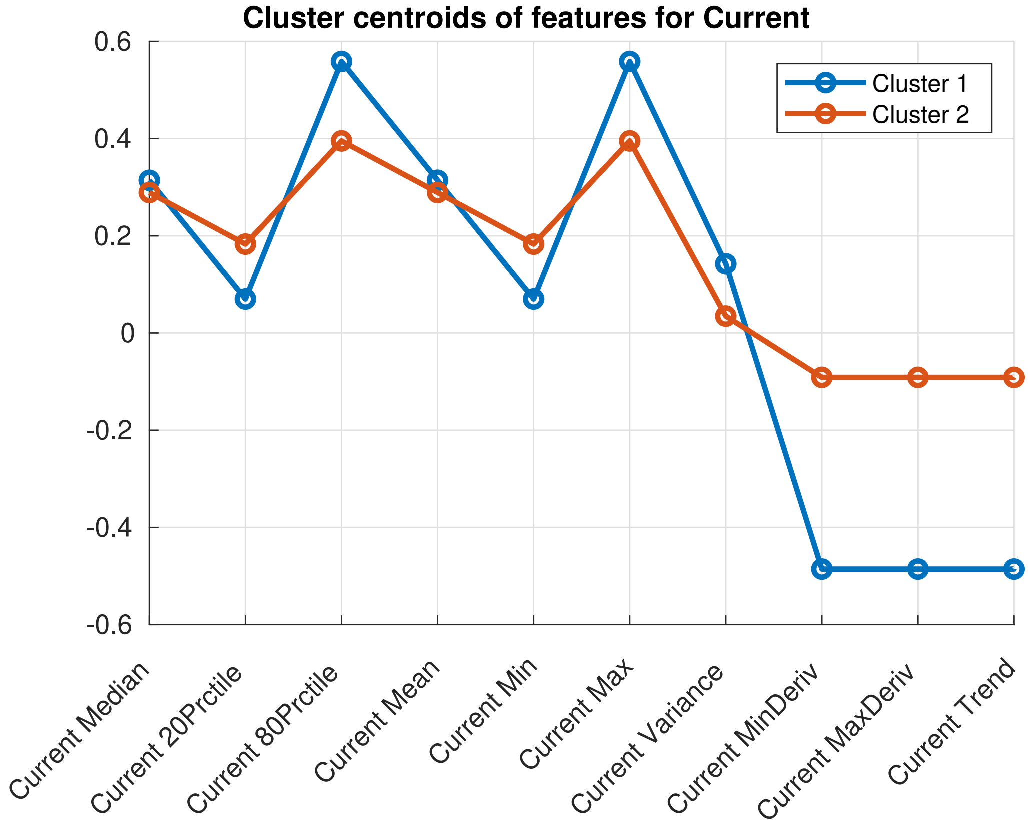

Figure 4.14: Centroids of the two clusters for windows of size TWO, in the features of the motor current data, found by the kmeans algorithm.

Performance metrics

The diagrams in figure 4.18 show the mean precision and recall scores of *riding* windows and *fall-off* windows, along with the 95% confidence interval bounds for every score. For the individual diagrams, as well as the exact scores for the metrics, see Appendix E.

Comparing the graphs across different window configurations, the scores are mostly similar. A tendency among all configurations is that the recall score for *fall-offs* are the lowest scores for most models, save for bagged trees. The bagged trees models, shown in purple, usually have the most evenly distributed scores. A recurring pattern for most models (except bagged trees in most cases) is that a high recall score for *riding* windows is usually accompanied by a lower recall score for *fall-off* windows. Most models also have a higher recall score than precision, or vice versa.

It is worth noting that, although not shown in the graph, both *idle* and *disembark* windows are rarely misclassified by the models. These window classes are defined by their strong characteristics, *idle* windows having very low motor current and eRPM and *disembark* windows having a strong negative trend. In essence, this means that the two window types that has to work the hardest to distinguish between are *riding* and *fall-off* windows. When the classifier mainly has to distinguish between two window types, the patterns mentioned above means that the classifier is lopsided toward the window type with the higher recall score.

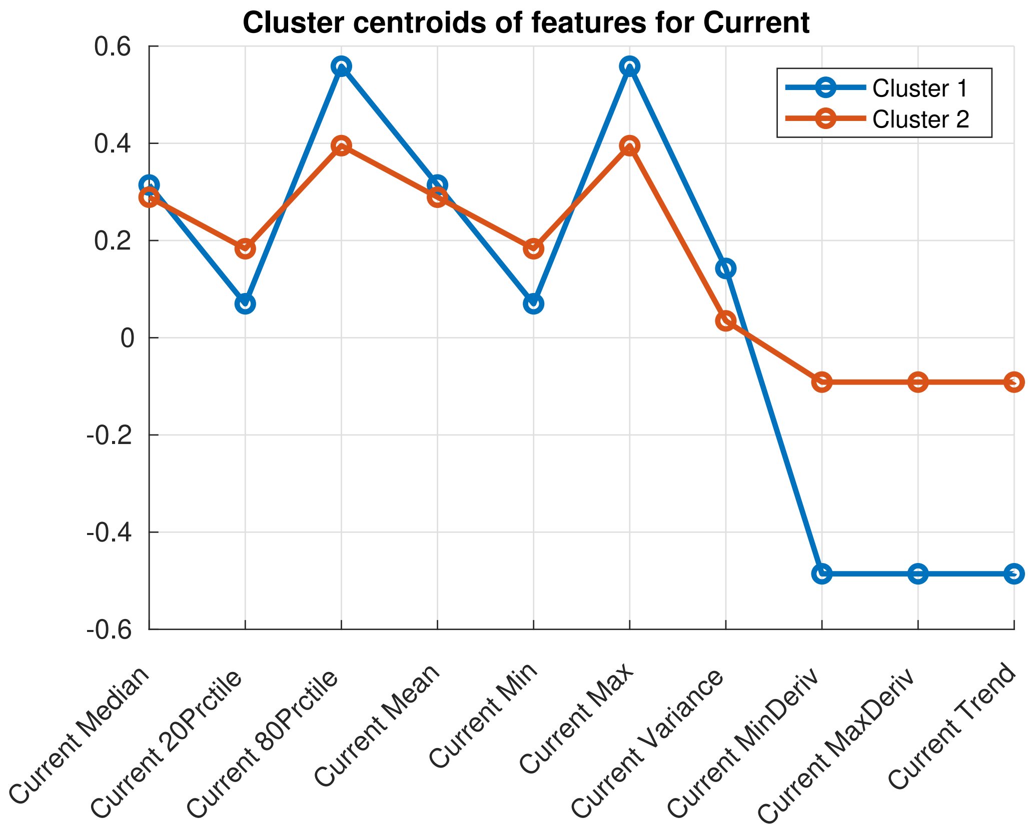

Figure 4.15: Centroids of the two clusters for windows of size TWO, in the features of the eRPM data, found by the kmeans algorithm.

The highest performance score for any model is circa 79,8%, which is the precision for *riding* windows using a CNN on windows of size 2. Otherwise, the majority of scores are between 65% and 78%. Taking into account all the performance metrics, the bagged trees model shows most promise, since its scores are more consistent, mostly ranging between 70% and 76%. Despite having the highest precision score, the CNN model for windows of size 2 had a very lopsided score, with a very low recall of 61,7 % for *fall-off* windows.

Visualization of classified windows

Due to both *disembark* windows and *idle* windows being defined by features in the motor data, it is important to see if the thresholds that were defined in section 4.7.4 actually improves the ability to distinguish between *fall-offs* and other window types.

In order to test the performance of a fall-off detection model, a manually generated ride log was downsampled to (roughly) 1 Hz. The original ride log is the same as the one seen in 4.8. The ride log was windowed and passed as input to a bagged tree classifier.

Figure 4.19 shows the same part of the ride as Figure 4.8. The red and green areas mark the windows that the classifier labeled as *fall-off* and *disembark*. Due to the overlap between windows, some areas of the graph can be considered as both *disembark* and *fall-off*. Since the *fall-off* areas are placed after the *disembarks* are placed, the green areas are sometimes colored over by red areas.

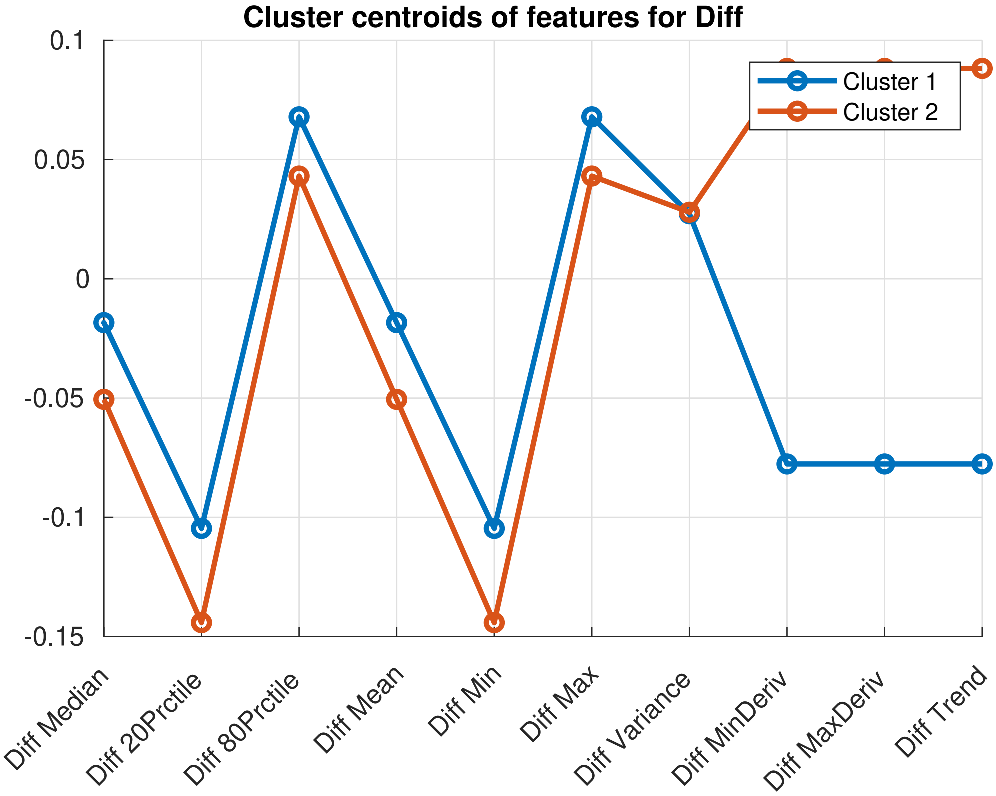

Figure 4.16: Centroids of the two clusters for windows of size TWO, in the features of the difference (eRPM - current), found by the kmeans algorithm.

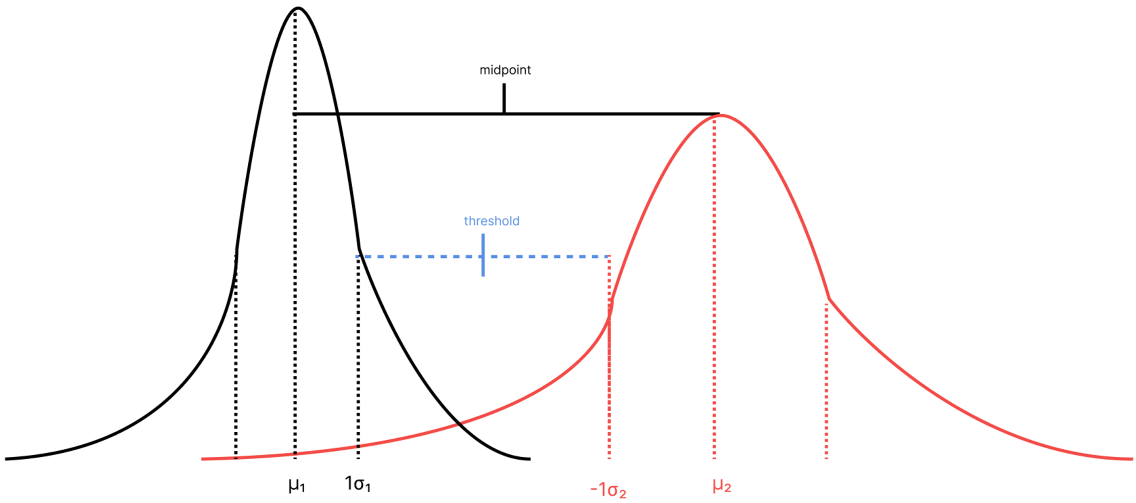

Figure 4.17: How the thresholds were set between *disembark* and *fall-off* windows, with respect to the distributions within each cluster.

The graph shows that the *disembark* classifications are found where there are almost complete stops, whereas the red areas seem to be found where there are downward or upwards trends in general. A wider look at this ride log can be seen in Figure 4.20. The *disembark*

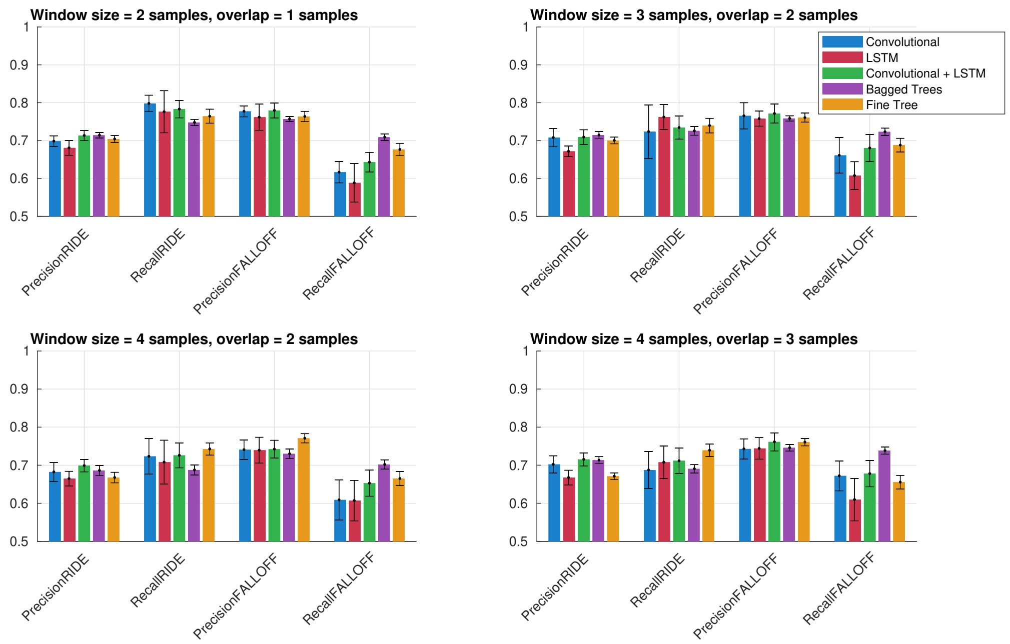

Figure 4.18: Model performances for different choices of sliding window sizes and overlaps.

classifications are in line with the limits defined in Section 4.7.4.

Figure 4.21 shows the same classifications but with windows of size two, with the threshold between *disembarks* and *fall-offs* as defined in section 4.7.4. The patterns here are mostly the same, with strong deceleration being classified as *disembark*. The *fall-off* classifications are less clear, but still seem to correlate with strong trends, but not stopping.

4.7.8 Takeaways from fall-off detection

From the plotted classifications in Figures 4.19-4.20 it can be seen that the current method of finding *fall-offs* specifically is severely lacking. The classifier is trained on data where the *fall-off* windows are when the DMS has disconnected, but the board is not already coming to a stop. There appears to be a great variety in how these windows look in the training data, because the *fall-offs* that are classified by the bagged tree model appear at times where the surfer evidently is not in process of falling off. This is also seen in Figure 4.18, the recall scores for the *fall-off* windows tend to be lower than all other scores.

The *disembark* classifications are more consistent, but still misclassify at times where there are no DMS disconnections. Even when correct, these classifications serve little purpose in a DMS algorithm, since the board is already stopping by the time a *disembark* is seen.

In Figure 4.18, the results are also shown for window size 4 with overlaps of 2 and 3. However, with these lengths of windows, the requirement for how fast the DMS should be able to react is not in compliance with the requirements discussed in section 4.1. They are still kept in the given results as they may be of interest to the reader.

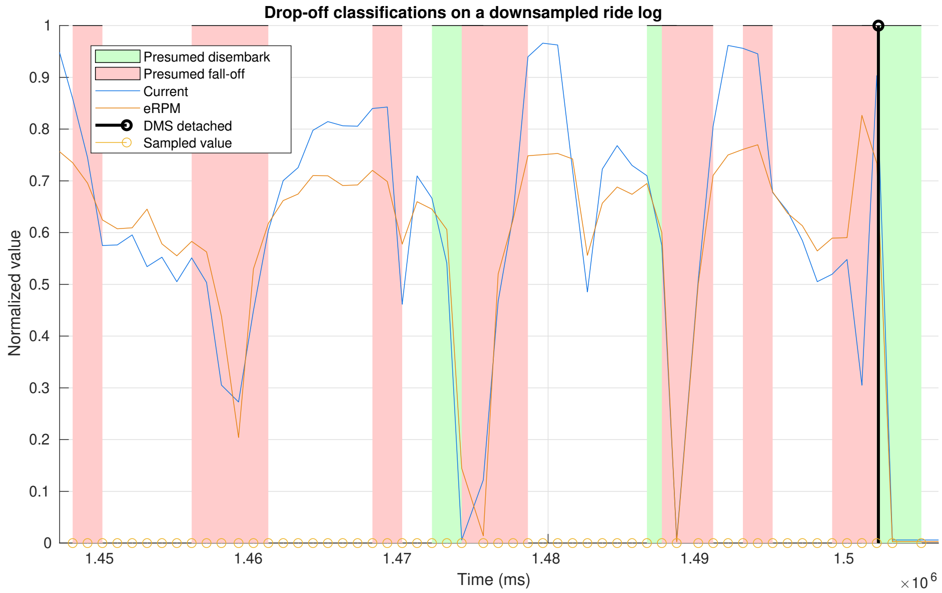

Figure 4.19: Visualization of disembark and fall-off classifications on a ride log, downsampled from 20 Hz to 1 Hz, split into 3-sample windows. The light red and light green areas show where the bagged tree classifier found a drop-off. The black vertical bar shows when the DMS was actually unplugged, meaning there was a fall off the board.

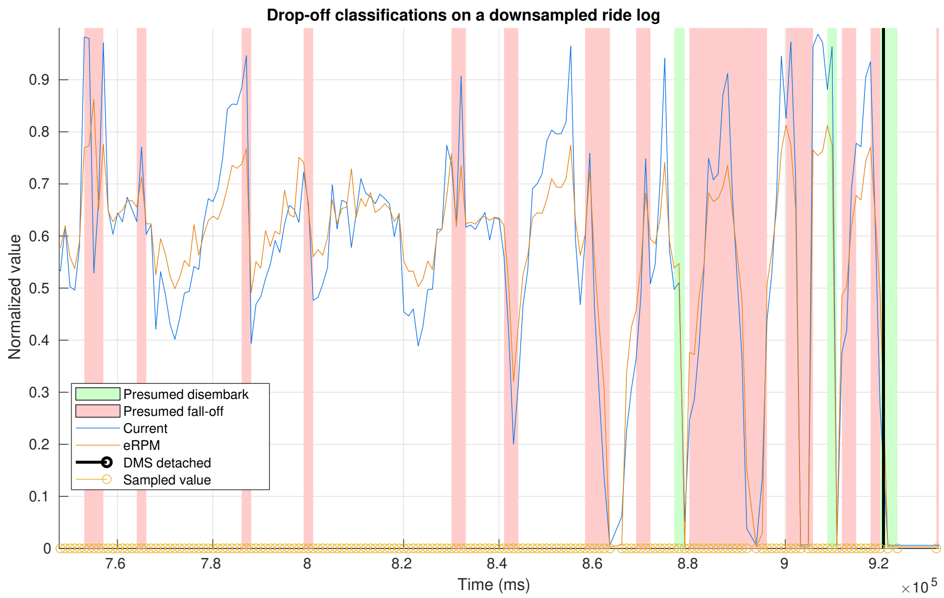

Figure 4.20: Another view of the same ride as Figure 4.19. Red areas mark windows classified as *presumed fall-offs*, green areas mark *presumed disembarks*. The yellow circles on the x-axis show the sample times.

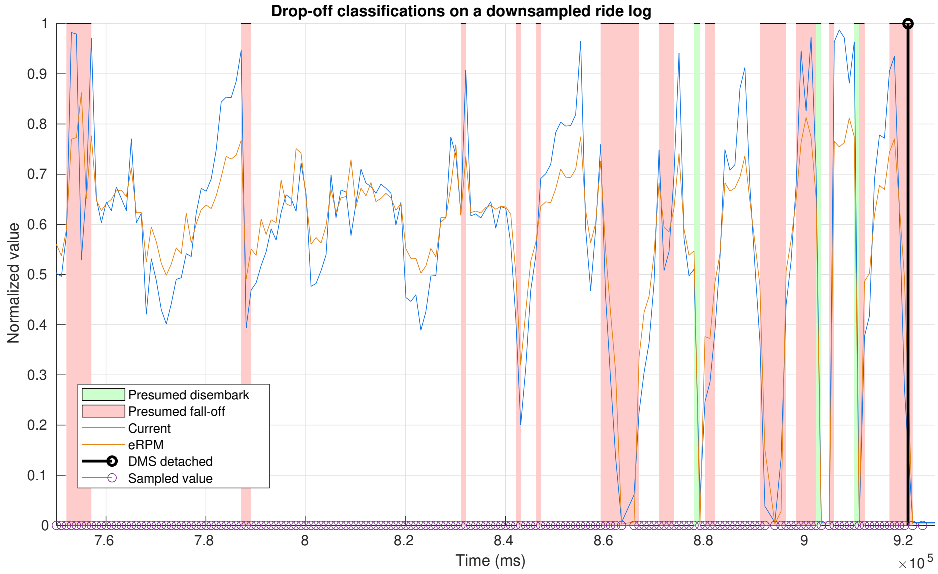

Figure 4.21: Visualization of disembark and fall-off classifications on a ride log, downsampled from 20 Hz to 1 Hz, split into 2-sample windows. The light red and light green areas show where the bagged tree classifier found a drop-off. The black vertical bar shows when the DMS was actually unplugged, meaning there was a fall off the board.

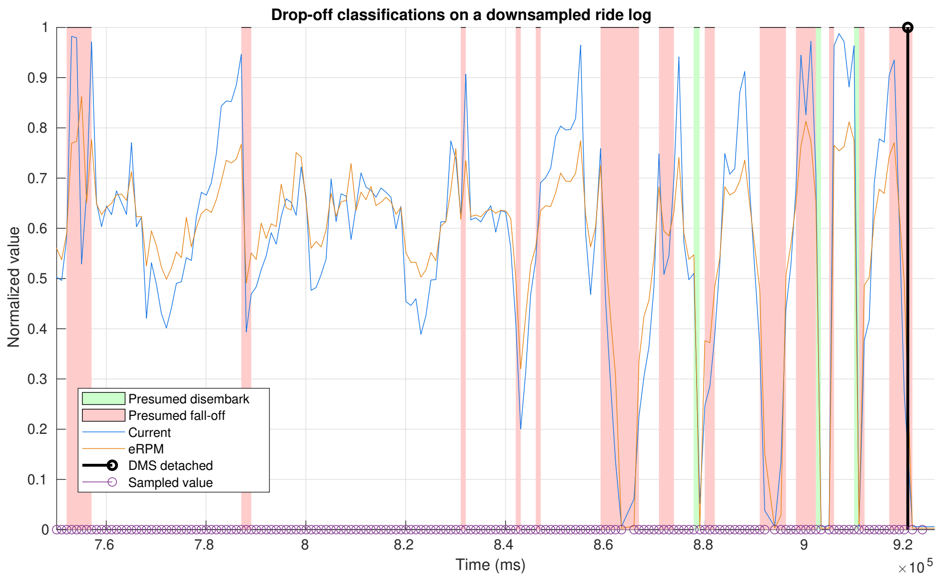

Figure 4.22: Another view of the same ride as Figure 4.21. Red areas mark windows classified as *presumed fall-offs*, green areas mark *presumed disembarks*. The circles on the x-axis show the sample times.

Chapter 5

Discussion

In this chapter, the methodology and results of the thesis will be discussed and critiqued. Discussions and critiques of the findings from evaluating the current DMS function will be presented, as well as discussions on utilizing the ML approach to potentially implement a similar function. The results will also link to the theories explored in the literature review, as well as to the three research questions. Potential areas of future work in the field will also be discussed.

5.1 Product analysis process

The methods chosen to evaluate the current DMS function succeed to indicate a motivation for a software-based implementation. The questionnaire and the interview questions were written to assess how the current leash-based DMS impacts the surfing experience in different ways. By evaluating the leash based on safety, convenience, comfort, functionality and aesthetics, the hope was to achieve rigidity in the analysis by data triangulation, to evaluate whether a software-based DMS could be an improvement in more aspects than one.

By conducting interviews with novice users as well as with more experienced users, nuance can be added to the analysis by looking at the differences and similarities between the two user groups. The results of the user study show that this evaluation method can be used to gather user sentiments regarding the DMS and to answer the research question. However, there were still some preexisting limitations and shortcomings in the study that could have been addressed to improve the evaluation. The following section will discuss these limitations and improvements.

5.1.1 User tests

This section looks at the process and results of the user tests and interviews. The participant selection process is discussed, as well as the choice of methods for data gathering.

Participants

The user tests and interviews were conducted during two occasions, with a total of seven participants that were all first-time users. The two distinct groups with common levels of experiences, the kite surfers and the non-surfers, made it possible to add another layer of nuance to the analysis by comparison between the two groups. However, all participants had existing relations with one or both of the researchers. This can be problematic from a research standpoint, as the previous relation could add bias to the answers of the interviewees. If the person had any knowledge about the purpose of the thesis, their answers could reflect this knowledge. They might have expressed thoughts and feelings that validate the research questions.

The selection process for participants in this study was subject to several influential factors. One such factor was that the study took place in the summertime during weekdays, which limits the pool of possible participants to people that are both not working and not on vacation. There was also an inability to reimburse the participants, which further limits the ability to recruit people. Finding people for the tests that had the possibility to participate with no pay, and the ability to come to Vombsjön by themselves, seemed a difficult task. Lastly, recruiting external participants was not advisable in this situation either, due to the expense of the equipment, a certain level of trust between the researchers and the testers was deemed beneficial.

The case can also be made that the study would be improved by more participants. Gathering more statements, thoughts and feelings regarding the leash would add more nuance and depth to the data, allowing for a more robust assessment of the leash and the tether-free DMS alternative. As previously mentioned, the ability to recruit users was limited by the time and the season of the tests, as well as the logistics of conducting them. A semi-structured approach was used due to its capacity to generate in-depth qualitative insights from each interview, while maintaining focus on the research topics. So, despite these limitations in the number of participants, the amount can still be deemed satisfactory for the purpose of the study.

Reactions from interviews

Conducting interviews immediately after the users' test runs provided some advantages. The inherently exhilarating experience of riding the board generated excitement in the participants upon their return to shore. This emotional state was somewhat present during the interviews. This could be attributed to two primary aspects: the thrill and enjoyment of riding or the frustration arising from difficulties in mounting the board. Additionally, the leash was often mentioned as a major hindrance during the mounting process, especially when it was inadvertently kicked off while trying to get on the board.

The influence of the leash on surfers with varying skill sets

Kicking off the magnet while climbing up the board exemplifies a problem potentially worsened by inexperience, as better board-mounting skills and enhanced balance, which reduces falls, are acquired with more experience. New users not only grapple more with the leash but also spend additional time in the water, where the leash has proven to be the most problematic, which can introduce further negative feelings towards the leash and make them more conscious of it. Evaluating the user tests and subsequent interviews somewhat strengthens this theory, as variations in participants' experiences with surfing activities indicate that those with less experience perceive the leash as a more dominant element of the surfing experience. Consequently, certain issues might be perceived as more pronounced for an inexperienced first-time user than for an intermediate or advanced rider. While the leash may present challenges for a first-time user, it may not significantly impact a typical user who has used the board a few times, due to the improvement in leash-use proficiency as one gains more experience.

5.1.2 Survey

In this section, the characteristics of the survey results are explored. The section underscores the value of a diverse participant group for robust findings. The survey methodology, aspects of self-assessment, and the precision in question wording are discussed. The emphasis is on clear questions and a detailed approach to understanding participant skill and experience to gather precise and relevant responses.

Participants

Similar to the interviews, the results from the survey could have been improved by an increased number of participants. The quantitative nature of the questionnaire lends itself well to gathering input from a large number of participants. More respondents would mean that the analysis could include finding different sub-groups within the responses, for example by analysing the differences between people that rated their level of experience on the board higher and lower.

Like with the interviews, the number of possible participants for the survey was limited by it being early July when the questionnaire was distributed, many people were out of office at that time. The responses from this part of the study were gathered by circulating the online questionnaire in Radinn's internal Slack channel. This presumably limited the pool of participants to employees of Radinn, which in itself can be a limitation to the evaluation. The surveys were anonymous, but if people are somehow involved in the development of the Radinn boards, it might be in their best interest to think about new innovations for the product, thus potentially having a skewed point of view when answering the survey. Subsequently, when answering questions regarding possible issues with the board, the respondent may perceive these issues as more severe. This perception is not only because it negatively impacts their own experience but also because it could potentially be detrimental to the product, thereby intensifying their negative feelings towards the issue. Therefore, a greater effort should have been made to increase the diversity of participants and recruit outside the pool of employees at Radinn.

Assessing skill levels

Out of the 13 participants in the survey, only one person assessed their skill level as below 7 out of 10. This can showcase some shortcomings in both the number of participants, as well as the design of the survey. While asking users to assess their experience level did show that the respondents were not complete beginners, the lack of variation in the answers to this question did show the potential disadvantage of using a single question to assess skill level. By also including questions about how long they have been jet surfing, how often they do so, or how many times in total they have surfed, a better picture about the skill distribution among the respondents could be painted. A high self-assessed skill level could mean that the person got the hang of the board quite easily to start, but still have not surfed much at all, meaning that any potential issues with the leash would only be experienced by the surfer this one time. Time spent on the surfboard is one aspect that is not captured by the current question. If someone who has surfed many times over a long period experiences the same issues every time, this would be useful information for the evaluation.

Question wording

Some improvements could also be made regarding the wording of the questions in the questionnaire. During analysis of the test answers, some questions were identified as potentially ambiguous. For instance, the respondents were asked to rate if they would rather use the board without the leash, even with increased risk for personal injury (see Appendix B, question 12). This question can be interpreted in two different ways, one way is to think of it as removing the DMS function completely, as if the magnet was connected without the leash, and the other would be that the leash-based DMS solution would be replaced by a tether-free alternative. Since the respondents are employees of Radinn, the purpose of the questionnaire might be known to some, and they might believe that the second alternative is what was implied. This ambiguity affects some parts of the analysis: a high rating (saying they would prefer to ride without a leash) can be seen as an outright approval of a tether-free DMS, or implied approval, but people that answered this question with a low number could have answered this way as a result of interpreting the question one way, but would have answered differently if they had interpreted the question the other way.

Answer scales

Most questions in the survey were all close-ended with answer alternatives being numbers on scales 1-5, 1-6 or 1-10. This was done to provide more answer alternatives than the Likert scale, and to avoid the users always selecting the neutral option. An issue with this approach is that the scales are not explicitly tied to a concrete answer alternative. Many of the questions asked for the frequency in which a user experienced certain issues. If you rate the frequency of an issue as a 3 on a scale of 1 to 6, it might not be as clear how frequent the issue is, in contrast to if the answer alternative had been a concretely defined frequency, such as "fewer than half of the times I surf".

5.1.3 Questionnaire and interview consistency

Greater care could have been taken to ensure that the evaluation of the leash in the interviews and the survey had been closer correlated. The questionnaire and interviews were intended to evaluate user sentiments regarding safety, convenience, comfort and functionality. However, upon reviewing and comparing the questions after the tests, there are parts of the interview that are not covered similarly in the survey and vice versa. For example, the survey asks if people would prefer to ride without a leash, even with a greater risk of personal injury (see Appendix B question 12). This is used in part to directly gauge interest in a tether-free DMS. The interview instead asks directly if they would like a tether-free alternative if the same functions were guaranteed (see Appendix A).

5.2 Machine learning modelling

It became evident throughout the process of this thesis that the machine learning work was more about processing data than it was about constructing the perfect ML models. *Idle* windows and *fall-off/disembark* windows were defined manually, through defining thresholds in the motor data. Finding a good threshold for these definitions is more important than finding a model that can place new data in these categories. If the categories are wrong to begin with, the classifier becomes fundamentally flawed despite its performance. The result from this thesis could possibly be improved in this regard by making some adjustments, ranging from the workflow to the pre-processing of data.

5.2.1 Data collection and pre-processing

In this section, techniques and challenges associated with preparing the data for analysis is addressed. The methods for constructing data windows, managing inconsistent samples, and the role of feature selection in influencing model performance are discussed.

Windowing of the data

An improvement in the data pre-processing would be to construct windows based on a time span, rather than a number of samples. The number of samples were chosen so as to approximately cover a specific time span, with the time between samples being calculated based on the window's time span and the number of samples in a window. Sometimes, the sampling time would become inconsistent or larger than expected or tolerated, mostly in instances directly following the DMS being disconnected. In the pre-processing stage, windows with inconsistent or large sampling times would be discarded. This could directly affect the performance in the fall-off detection task, since what happens directly after the DMS is triggered could be useful data. Splitting into windows based on time spans would mean that windows are not removed based on this criterion, but would result in some windows containing less data than others. Those windows would need extra care when the time series data is used in the LSTM models, due to the new windows being of variable length, but the features used for the other models could be calculated the same way.

Faulty samples

It was found during the thesis that the sampling of the motor data was not always flawless. Occasionally, a random sample of eRPM or motor current would be NaN (Not a Number), where it was otherwise obvious that the board was being used. These samples are few and far inbetween, but still have an impact on the features of a data window. This would impact both the minimum value of the metric within that window, as well as influencing the statistical metrics such as the mean and median. This problem could have been fixed simply by a script comparing a suspected faulty sample against the surrounding samples. If the sudden dip is too large to feasibly happen during regular usage of the board, it could be corrected to another more reasonable value given the context.

Deciding features

A big part of enabling oneself to find patterns in data is to find the correct features to represent the phenomena. All the features for this thesis were derived from the motor current values, the eRPM, as well as the difference between the two. For the fall-off detection task, the normalized time series values were used, while the on-board detection task used a total of 21 features, seven features for each metric. While these calculated features can represent the distribution of the values inside a window, there might still be features that were missed that could have improved the model. Expanding this initial feature array to include more features would serve as a better base for the feature selection that was conducted later.

Refining of the feature selection was done at a stage of the thesis where the window definitions, feature arrays, and model definitions were thought to be good enough. The purpose was to reduce redundancy and make the classification more efficient, without affecting any model's performance negatively. However, the work persisted for some time after this, with adjustments being made in some of the previously mentioned categories. The chosen features stayed the same after the feature selection stage, which might have been detrimental to the performance of subsequent models.

5.2.2 On-board detection

This subsection describes the manual collection of data for the task, highlighting the challenges encountered. It also delves into how varying sampling frequencies could influence model results and concludes by evaluating the ML models' ability to differentiate between manned and unmanned rides.

Comments on the performance

The results from the exploratory data analysis seemed promising, showing that there was a slight observable difference in the motor data depending on if the board was manned or not. However, the results show that the ML models cannot distinguish on-board and off-board windows at a high enough accuracy to be reliable as a DMS alternative. Like the EDA shows (see Figure 4.6), there are some differences in the distributions of the features between the window types, but there are also overlaps between the two sets (as also seen in Fig. 4.6). Improvements could be made by looking at and comparing specific kinds of windows, such as acceleration and deceleration windows, to compare these scenarios between on-board and off-board windows.

Data collection

The data required to perform the on-board detection task had to be gathered manually, one person had to control both the surfer-less board and their own board simultaneously. This was an impractical procedure, as the surfer-less board could only be used when the controller was in close range. It was also logistically difficult to arrange the data collection session due to the equipment that was needed and the two-board setup was also difficult to use. This limited the ability to gather versatile off-board data.

Due to the quick acceleration and high speed of a surfer-less board, staying in range became a difficult task. The result was that most data generated was shorter spurts of quick acceleration, followed by the board running to a halt once the controller was out of range. This means that the *off-board* windows have an over-representation of acceleration and deceleration windows, which in turn impacts what the model believes an *off-board* data window looks like. This effect can be seen in Figure 4.8 and 4.9. An intended fix was to gather off-board data a second time, taking greater care to gather more diverse data. However, the skewness is still evident in the models. Another possible improvement here would be to compare strictly between different scenarios, as described in the previous subsection.

Comparing performance for different sampling frequencies

An initial idea was to investigate how the sampling frequency affects the performance of the models. Downsampling the 20 Hz data to lower frequencies and running the same models on the downsampled data would have shown this effect. This idea could not be pursued in the end however, simply due to lack of time.

5.2.3 Fall-off detection

This section delves into the methods and challenges faced when addressing fall-off detection. It emphasizes the challenges posed by the chosen sampling frequency, the significance of data resolution in capturing short events and patterns, hence, emphasizing potential advantages of higher sampling rates. The section also discusses the practical aspects of data collection and provides insights into the model's performance.

Discussing the performance

The models were mostly detecting the trends of the windows, as shown in Figure 4.20-4.21. The *disembark* classifications were made when the board had come almost to a stop. Due to the thresholds defined between *disembarks* and *fall-offs*, there is a greater variety in what can happen within the context of a *fall-off* window. A similar amount of variety is possible for a *riding* window, since the only limitation is that it cannot be *idle* (very low motor activity) and that the DMS is not disconnected. Since the *fall-off* windows in their current definition do not exhibit a unique enough pattern, it makes sense that these windows and the *riding* windows are the ones most frequently misclassified as one another.

Data resolution

The sampling frequency of 1 Hz made it challenging to detect short-spanning activities such as falling off the board. Looking at figures 4.19 - 4.22 it can be seen that patterns that the models were trained to associate with fall-offs were present all throughout the ride log. For data windows of size three, it was also possible for a window to "dip" and come back up again and still be classified as a *fall-off*. A longer window span seems to come with too much uncertainty, while the shorter windows have less data to use in the decisions. Even if the data was fully annotated, difficulties could still arise due to the nature of measuring events only once per second, especially for dynamic activities like riding a jetboard, with the potential for rapid sequences of events.

Collection of data

Utilizing the data available in the database was necessary because manually collecting fall-off data at a higher resolution was considered impractical, risky and logistically difficult. Additionally, when multiple rides were performed and numerous orchestrated falls were generated, there was no certainty that these falls accurately represented real fall-off activities, given that they would be performed and not "natural". This could become problematic when it comes to generalizing a model trained on this "artificial" fall-off data.

Raising the sampling frequency

As discussed previously in this section, a sampling frequency of 1 Hz makes it difficult for any advanced level of pattern recognition. Although it is uncertain whether there are patterns to be found that indicate fall-offs, the possibility of finding out this information would likely increase with higher frequency data. One key reason for this improvement with higher resolution is that data points collected closer in time exhibit stronger relationships, enabling the detection of subtle changes, as opposed to only capturing major shifts when there is a significant time gap between data points. Some potential fall-off data can be extracted from the data gathered for the on-board detection task, but the number of windows that this generates is small since not many fall-offs were attempted. The task of fall-off detection on higher frequency data remains interesting and still mostly unexplored. A benefit of investigating fall-off data is that the DMS data makes it possible to extract potential fall-off data without going surfing yourself. This data could therefore potentially be acquired by raising the sampling frequency on Radinn's fleet of boards.

5.3 Answers to research questions

5.3.1 RQ1: How is the user experience affected by the current DMS system?

The user study shows the importance of the DMS function. It is reassuring for some users that you need to connect the magnet to the board for it to run, to ensure that the board cannot be accidentally launched. The physical leash is an integral part of the current DMS solution. For the board to work, it is necessary for the user to tether themselves to the board using the leash.

As shown by the study, having to be tethered to the board to use it results in some inconvenience for the surfer. Across all user groups, the leash and magnet causes issues, particularly when attempting to mount the board. The frequency of problems when climbing onto the board was generally high among experienced users, and many interviewees mentioned accidentally kicking off the magnet while mounting the board as well. This is an inconvenience that can be frustrating, but is ultimately not a risk to the user. However, when the magnet is dislodged while surfing, the result is a personal risk of being hurt and an annoyance. First-time users and experienced users alike mentioned the magnet accidentally dislodging while surfing. Among the experienced users, a majority answered that they relatively frequently have to deal with the leash becoming entangled in them or the board, which is both uncomfortable and may cause the magnet to disconnect.

Although experienced users expressed more frustration over the limitations of the leash, beginner users might be the ones most affected by it. Issues like the leash dragging in the water, or accidentally kicking off the magnet when mounting the board, are issues that disproportionally affect the users that have not yet become familiar with the equipment or how to ride the board. The leash dragging in the water can be mitigated in part by attaching the leash higher up on the leg, which not many users do. Becoming better at jet surfing means that you are less prone to falling off, resulting in less time having to deal with the leash when mounting the board, or when in the water.

The aesthetic impact of the leash is largely negligible, with few users claiming that it has an effect on how they perceive the board. There were a total of three users that claimed it has any impact whatsoever, but the vast majority of the participants in the study remained neutral. As the DMS feature is perceived to be important by the participants, it seems that the appearance of the leash is secondary to the function it fulfills.

Despite the inconveniences caused by the leash, there is a degree of safety and reassurance that is added by its presence. The connection serves as a representation of the state of the system: if you are not connected to the board, you can not use it. This reassurance is not desired by everyone, however, as some users from both test groups expressed a desire to ride without the leash. There seems to be a user group that do not appreciate the limitations that they feel the leash and magnet impose.

Overall, the current DMS solution reliably fulfills its function, to cut the driveline when a user falls off. It provides a sense of security due to its simple and reliable implementation. However, it is not perfect, as it can be both inconvenient and uncomfortable to use the leash and magnet system. The leash can become entangled in the surfer's body or the board, causing discomfort and inconvenience, as these issues have to be dealt with. Additionally, these issues affect all users, but could be experienced more by those that tend to fall off the board, adding more inconvenience to an already frustrating situation.

5.3.2 RQ2: What are the motivations for implementing a tether-free DMS system?

The motivations for implementing a tether-free DMS solution can mostly be derived from the issues described in the discussion of the previous question. The removal of the leash, the magnet, and the need to tether yourself to the board would address most of the inconveniences described by users. A tether-free solution could be an alternative for users that have felt more affected by the leash, or a complement to the physical DMS that might work better in certain scenarios.

The overarching motivation behind implementing a tether-free DMS system would be to address the inconveniences caused by the leash and the magnet. There would simply not be any magnet to dislodge, or leash to become entangled in. It would provide a greater freedom of movement both on the board and in the water, which is desired by some users. The removal of the leash could potentially improve the general user experience of the board if safety aspects and other functions, highlighted as important by test users, could be covered by the software based solution.

The tether-free DMS would be an addition, or an alternative, not a complete replacement of the current DMS solution. It would be unwise to disregard the current version regardless of how well the software-based solution works, since some users expressed that the reliability and simplicity of the physical leash is favorable. For users that wished to do so, a working tether-free mode should be available for riding more freely, but without encroaching on the other users' wish to ride with an increased sense of safety.

5.3.3 RQ3: How well can rider presence be detected with machine learning, utilizing data from the ride logs?

The results of the fall-off detection and on-board detection tasks show that distinguishing between rider presence or absence can be done quite well, but not nearly well enough to fulfill the requirements of a DMS feature, as outlined in section 4.1. Although there are differences between off-board windows and on-board windows, as well as fall-off windows and riding windows, there are similarities that make perfect classification difficult. All the models had varying performances depending on window types. Figure 4.7 shows that the off-board windows could be correctly classified up to 92,6% of the time, but the on-board windows were only classified correctly at most 77,7% of the time. The fall-off detection models classified riding windows correctly about 72,4-79,8% of the time, but had recall scores at about 58,8-72,3% for the fall-off windows. These scores show that there are some windows that can be distinguished from each other, but not all. The graphs in figures 4.8-4.9 and 4.19-4.22 show that the skewness of the models would result in falsely classified rider absence. The models by themselves cannot detect rider presence at a sufficient level to fulfill the purpose of a DMS.

Based on the findings of the fall-off detection task, a software-based DMS solution is not possible with the current sampling frequency of 1 Hz. There is too much uncertainty in the data and the time spans of the data windows are too long to meaningfully distinguish between how it looks when someone is riding the board normally and when they are about to fall off.

On-board detection is an idea that could be refined. There are fine tunings that could be made to the approach to better distinguish between the on-board and off-board scenarios, such as comparing instances where someone is starting or stopping. However, the performances shown in this thesis are not good enough to fulfill the requirements of a DMS function either.

5.4 Future work

Throughout this thesis, numerous discussions were had concerning potential future implementations and alternative work extending from the work in this thesis. This can be to explore ways of implementing a DMS algorithm, to including other types of data in the dataset. To actually implement a software-based DMS on a board, more work needs to be done, but an implemented prototype would open the possibility of conducting more studies on the most user friendly design of such a feature.

5.4.1 Algorithm

Classifiers are seldom perfect, which is at odds with the requirements for a DMS feature. In a real-life scenario, with a software-based DMS implemented on a board, stopping the driveline as soon as a single window is classified as fall-off/off-board would be unwise, since a classifier is unlikely to always predict the right class and due to the risks associated with stopping the board abruptly. In order to increase robustness in a software-based DMS solution, an algorithm that utilizes the strength of predictions and aggregates several classifications could be implemented. Most classifiers generate a probability for each class, selecting the most probable class as its final output. Utilizing the probability output, if the strongest classification by the model has a low probability, it can be compared to surrounding classifications and changed to a more probable one. By aggregating predictions and only triggering the DMS after a set number of off-board/fall-off classifications, performance could be improved.

5.4.2 Utilizing additional sensors

It was clear from observing the board that an unmanned board's acceleration during a launch varies significantly depending on whether there is a rider present or not. Including sensor data such as accelerometer readings from both the controller and potentially the board, would be valuable to assess. The relationship between the current, eRPM data, and accelerator data from the controller could be analyzed to determine how the rider's acceleration correlates with the eRPM and current. Furthermore, incorporating data from an onboard accelerometer can help verify if the rider and board are accelerating in sync. Additionally, combining sensor data with motor data may provide a more precise detection of fall-off incidents than using motor data alone.

5.4.3 DMS design

In this thesis, an evaluation was conducted only regarding the *removal* of the magnetically-corded leash. Future work could involve refining the design of the current solution, exploring alternative magnet-cord-based designs, or seeking entirely different approaches altogether. Additionally, results from the user tests emphasize that visibility regarding whether the board is armed is indicated by whether the leash is connected in the current leash solution. If the software-based solution is implemented, a future task could be to design and evaluate a functionality to indicate if the board is armed. However, this indicator should ideally only use hardware already present on the board.

5.4.4 Increasing sampling rate for fall-off detector

In this thesis, one of the constraints was the use of only low-frequency data for the fall-off detector. This limitation arose because Radinn's database did not contain higher frequency data, and collecting such data was not feasible within the scope of this study. However, should such data become available in the future, it would be intriguing to conduct a similar evaluation as done in this thesis, in order to investigate the results yielded from higher resolution.

Chapter 6

Conclusion

The objectives of this thesis were to assess how the user experience is impacted by the current DMS solution, to determine the motivational factors for considering alternatives to the current method of detecting a rider's presence and to explore the possibility of implementing a software-based DMS solution, utilizing data from ride logs. The first two objectives were investigated through a user study consisting of tests, interviews and a survey, while the last objective was investigated through ML modelling.

The user study evaluated how the leash-based DMS impacted user experience based on safety, convenience, reliability and aesthetics. The findings from the user study suggest that a tether-free solution might enhance the user experience for certain users. Some users appreciate the simplicity and reliability of the current DMS, that the concept of the magnet and the leash is easy to grasp and provides safety in knowing exactly whether a board is usable or not. Inexperienced and experienced users alike can experience the leash as inconvenient, many people from both groups experience problems with the magnet accidentally detaching while surfing, or entangled in the board or the body. The study showed that the DMS feature is important for a jet surfboard, but that the current implementation negatively impacts the user experience for many people. Some users also expressed outright the desire to have a tether-free DMS feature. Not every surfer would benefit from or opt to use a tether-free solution, but if such a solution could match the precision of the existing leash-oriented method and include features like indicating the board's armed status, it might be universally appreciated.

From an initial data analysis, it was shown that there were some differences between *on-board* and *off-board* data. The subsequent ML modelling showed that there is a difference, but that there also may be overlaps between the two classes that make the models classification attempts fall short. The recall scores for *on-board* and *off-board* windows ranged between 70.2-77,7% and 84,8-92,6%, respectively. Improvements could be made by creating a more diverse set of *off-board* data, or by specifically comparing how acceleration or deceleration looks for the two types of data. Ultimately, no on-board detection models perform well enough in their current form for DMS purposes.

The fall-off detection methods also fell short, with recall scores for *riding* windows around 72,4% to 79,8%, and *fall-off* between 58,8% and 72,3%. A limitation for this task was that the data was sampled at a frequency of 1 Hz. Increasing the sample rate could improve accuracy and responsiveness, by enabling the detection of finer patterns in the data. Attempts were made at defining a threshold between what was seen as a *fall-off* and a *controlled disembark*. It was shown that windows classified as *fall-offs* many times overlapped with *riding* windows. Windows classified as *disembarks* were ones where the board appeared to stop, which still resulted in some misclassifications. Ultimately, it was shown that attempting to find patterns that indicate a fall-off is not possible with low frequency data.

An implementation of an accurate ML-driven DMS model could be applied to other PWCs or to other machinery that could pose a danger if ran without an operator. Not only could it enhance the user experience of a product by eliminating a physical DMS like a leash, button, switch, or similar, but it could also serve as an additional security check, ensuring the system does not solely rely on mechanisms which could potentially be overridden or not used.

References

[1] Wärtsilä Netherlands B.V. Midsize waterjets product guide, 2023. URL: https://www.wartsila.com/docs/default-source/product-files/gears-propulsors/waterjets/product-guide-o-p-midsize-waterjet.pdf.

[2] François Chollet. *Deep learning with Python*. Manning Publications, 2017.

[3] Marc Peter Deisenroth, Aldo A. Faisal, and Cheng Soon Ong. *Mathematics for Machine Learning*. Cambridge University Press, 2020.

[4] Lars Torsten Eriksson and Finn Wiedersheim-Paul. *Att utreda, forska och rapportera*. Liber, 2014.

[5] BoatUS Foundation. Personal water craft, 2023. URL: https://www.boatus.org/study-guide/activities/pwc/.

[6] Daniel Frotan and Jens Moths. Human body presence detection in water environments using pulse coherent radar, May 2021.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org.

[8] Austin Hughes. *Electric Motors and Drives: Fundamentals, Types and Applications*. Newnes, 2019.

[9] IBM. What are neural networks?, 2023. URL: https://www.ibm.com/topics/neural-networks.

[10] Lazar Jonathan, Feng Jinjuan Heidi, and Hochheiser Harry. *Research Methods in Human Computer Interaction (Second Edition)*. Morgan Kaufmann Publishers, 2017.

[11] Raman Kumar and Anuj Jain. Driving behavior analysis and classification by vehicle obd data using machine learning. *The Journal of Supercomputing*, pages 1–20, 2023.

[12] Byung Il Kwak, JiYoung Woo, and Huy Kang Kim. Know your master: Driver profiling-based anti-theft method. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pages 211–218, 2016. doi:10.1109/PST.2016.7906929.

[13] Song-Mi Lee, Sang Min Yoon, and Heeryon Cho. Human activity recognition from accelerometer data using convolutional neural network. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 131–134, 2017. doi:10.1109/BIGCOMP.2017.7881728.

[14] D.R. Lenorovitz, E.W. Karnes, and B. Haygood. Personal watercraft (pwc) injury hazards–analyses, technical advancements, and continuing safety challenges. *Theoretical Issues in Ergonomics Science*, 21(3):285–311, 2020.

[15] Rozum Robotics LLC. Brushless motor, 2023. URL: https://rozum.com/brushless-motors/.

[16] MathWorks. K-means clustering, n.d. [Accessed: 2023-10-18]. URL: https://se.mathworks.com/help/stats/kmeans.html#buefs04-3.

[17] D. Morice, M.A.R. Stokes, J. Davey, and S. Skinner. Severe anorectal injury secondary to jet ski accident: An important and increasing mechanism of injury. *Trauma Case Reports*, 2022.

[18] Steven Morris. Cornwall speedboat accident report says 'kill cord' not attached to driver. *The Guardian*, May 17 2013. URL: https://www.theguardian.com/world/2013/may/17/cornwall-speedboat-accident-kill-cord.

[19] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[20] United Nations. The sustainable development goals, 2023. URL: https://sdgs.un.org/goals.

[21] S. Gopal Krishna Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *CoRR*, abs/1503.06462, 2015. URL: http://arxiv.org/abs/1503.06462.

[22] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005. doi:10.1109/TPAMI.2005.159.

[23] Jennifer Preece, Helen Sharp, and Yvonne Rogers. *Interaction design: beyond human-computer interaction*. Wiley, 2015.

[24] Andy Priestner. *A handbook of user experience research and design in libraries*. 2021.

[25] Radinn. Radinn electric surfboards, 2023. Accessed: 2023. URL: https://www.radinn.com/.

[26] Louis M. Rea and Richard A. Parker. *Designing and conducting survey research. a comprehensive guide*. Jossey-Bass, 2014.

[27] Royal Yachting Association (RYA). Use your kill cord, 2013. URL: https://web.archive.org/web/20130117003954/http://www.rya.org.uk/cruising/safety/Pages/Useyourkillcord.aspx.

[28] Bengt Starrin and Per-Gunnar Svensson. *Kvalitativ metod och vetenskapsteori*. Studentlitteratur, 1994.

[29] Monolithic Power Systems. Brushless vs brushed dc motors: When and why to choose one over the other, 2022. URL: https://www.monolithicpower.com/en/brushless-vs-brushed-dc-motors.

[30] Tom Tullis and Bill Albert. *Measuring the user experience: Collecting, analyzing, and presenting UX metrics*. Elsevier/Morgan Kaufmann, 2023.

Appendices

Appendix A

Interview questions

Intervjufrågor

Den här intervjun för vi med personer efter att de har åkt på en Radinn-bräda för första gången, snart efter att de har gjort sitt åk. Intervjun är semistrukturerad, så den kanske inte följer mallen helt och hållet, men tanken är att varje fråga ska beröras under intervjuns gång. Vi försöker få en bild av personens relevanta brädsportserfarenhet samt hur åket gick, för att sedan på leashen.

—

Innan åket:

- Åket sker på egen risk.
- Håll dig gärna en bit ut på vattnet. Dels för att få egen , dels för att inte råka få in sand och sten i motorn genom insuget. Börja köra först när vattnet är över knähöjd och du är säker på att inga är i vägen.
- Det är mycket kraft i motorn, börja med att köra lätt, tips.
- Tryck på gasen tills kontrollen vibrerar.
- Spänn fast kontroll och leash.
- Det är fritt att avbryta när man vill.

Innan intervjun:

- Intervjun och testet är en del av vårt examensarbete på Radinn och Designcentrum på LTH.
- Vi vill göra en utvärdering av behovet av leashen.
- Dina svar hjälper oss att få en bild av hur förstagångsåkare upplever sin åktur, samt deras upplevelse av leashen (den rem som går mellan åkarens vad och DMS:en på brädan).
- Syftet är inte att utvärdera dig som person eller hur du presterade. Vi är intresserade av dina upplevelser men det är leashen som vi undersöker.
- Jag kommer att spela in intervjun på mobilen samtidigt som jag för anteckningar. Vi kanske transkriberar delar av intervjun, men all din personliga information är anonym i det slutgiltiga arbetet, utom viss demografisk information (ålder, kön), om det anses behövas. Du kan fortfarande tacka nej till intervjun, eller välja att avbryta när som helst.
- Innan vi börjar kan du gärna *skriva på här* för att samtycka till databehandlingen. Allt inspelat material raderas efter vår analys.

Bakgrundsinformation

Hur gammal är du?

Vilket kön identifierar du dig som?

Tidigare brädsportserfarenheter

Har du sysslat med brädsport tidigare? (t.ex. Surfing (kite, wind, wave, wake, elektrisk), snowboard, skateboard/longboarding)

Är det något du gör regelbundet? *(Med fokus på den sport som är mest relevant eller personen har mest erfarenhet inom)*

- *Svarsalternativ 1, regelbundet:*
  - Hur ofta gör du det? Hur många gånger, på ett ungefär?
- *Svarsalternativ 2, enstaka tillfällen:*
  - Hur många gånger har du gjort det på ett ungefär?
- *Svarsalternativ 3, inte alls:*
  - Lämna ämnet.

Skulle du säga att du är väl erfaren inom sporten?

Skulle du säga att du är bra på det?

Tror du att det gjorde dig bättre på att åka den elektriska surfbrädan? Kände du att du kunde ta ut svängarna efter ett tag/direkt?

Åkupplevelsen

Hur gick ditt åk?

Vad tyckte du fungerade särskilt bra?

Vad tyckte du gick sämre?

Komfort

Var leashen i vägen någon gång när du skulle klättra upp på brädan?

Var leashen i vägen någon gång medan du åkte?

Kände du dig hindrad av leashen när du åkte, i flexibilitet och rörelsefrihet?

Satt leashen bekvämt på kroppen?

Upplevd säkerhet

Kände du att leashen hade en positiv inverkan på din känsla av säkerhet när du åkte?

Skulle du säga att leashen hade en positiv inverkan på din känsla av din omgivnings säkerhet?

Känner du dig säkrare med vetskapen om att leashen finns?

Övrigt / funktionalitet / utseende

Tycker du att det är en fördel att kunna se att brädan är körbar med hjälp av leashen (inkopplad)? '

Hade det varit lika betryggande om du visste att samma funktionalitet fortfarande skulle finnas, även utan leashen? Har den fysiska leashen i sig någon inverkan.

Hur tycker du att leashen påverkar brädans estetiska intryck? Samt hur du ser ut när du åker.

Tycker du att det finns några andra fördelar med den fysiska leashen?

I det stora hela, hur påverkade leashen hur nöjd du blev med ditt åk?

Några andra tankar, gällande åket i allmänhet eller leashen i synnerhet?

Appendix B

Questionnaire

1. **Have you ever used a Radinn electric surfboard?**
   Y/N
2. **Age:**
3. **Gender:**
   M/F/O
4. **Please rate your level of experience in riding a Radinn Electric Surfboard:**
   Beginner(1)/Expert(10)
5. **Have you had problems with the leash becoming entangled in your legs, arms, or other objects?**
   Never(1)/Often(6)
6. **Have you had problems with leash snap-back, when it is stretched and suddenly released?**
   Never(1)/Often(6)
7. **Have you personally encountered additional safety risks caused by the leash, beyond those previously mentioned? If yes, please specify:**
8. **Have you found the leash to be disruptive during surfing?**
   Never(1)/Often(6)
9. **Does the leash inhibit your freedom of movement and flexibility during surfing?**
   Never(1)/Often(6)
10. **Have you found the leash to be problematic when attempting to get up on the board from the water?**
    Never(1)/Often(6)
11. **Does the leash inhibit your freedom of movement and flexibility, when you are in the water and on the beach?**
    Never(1)/Often(6)
12. **Would you prefer not to use the leash while surfing, even with a slightly increased risk of personal harm?**
    Definitely not(1)/Absolutely(6)
13. **How does the leash impact the overall aesthetic impression of the board, in your opinion?**
    Greatly decreases(1)/Greatly Improves(5)

14. **How does the leash affect your overall satisfaction while riding?**
    Greatly decreases(1)/Greatly Improves(5)
15. **Do you see any advantages of being physically connected to the board, besides the function of shutting down the motor upon disconnection? If yes, please specify:**
16. **If there are any specific thoughts or concerns you would like to express, please share:**

Appendix C

Figures from exploratory data analysis

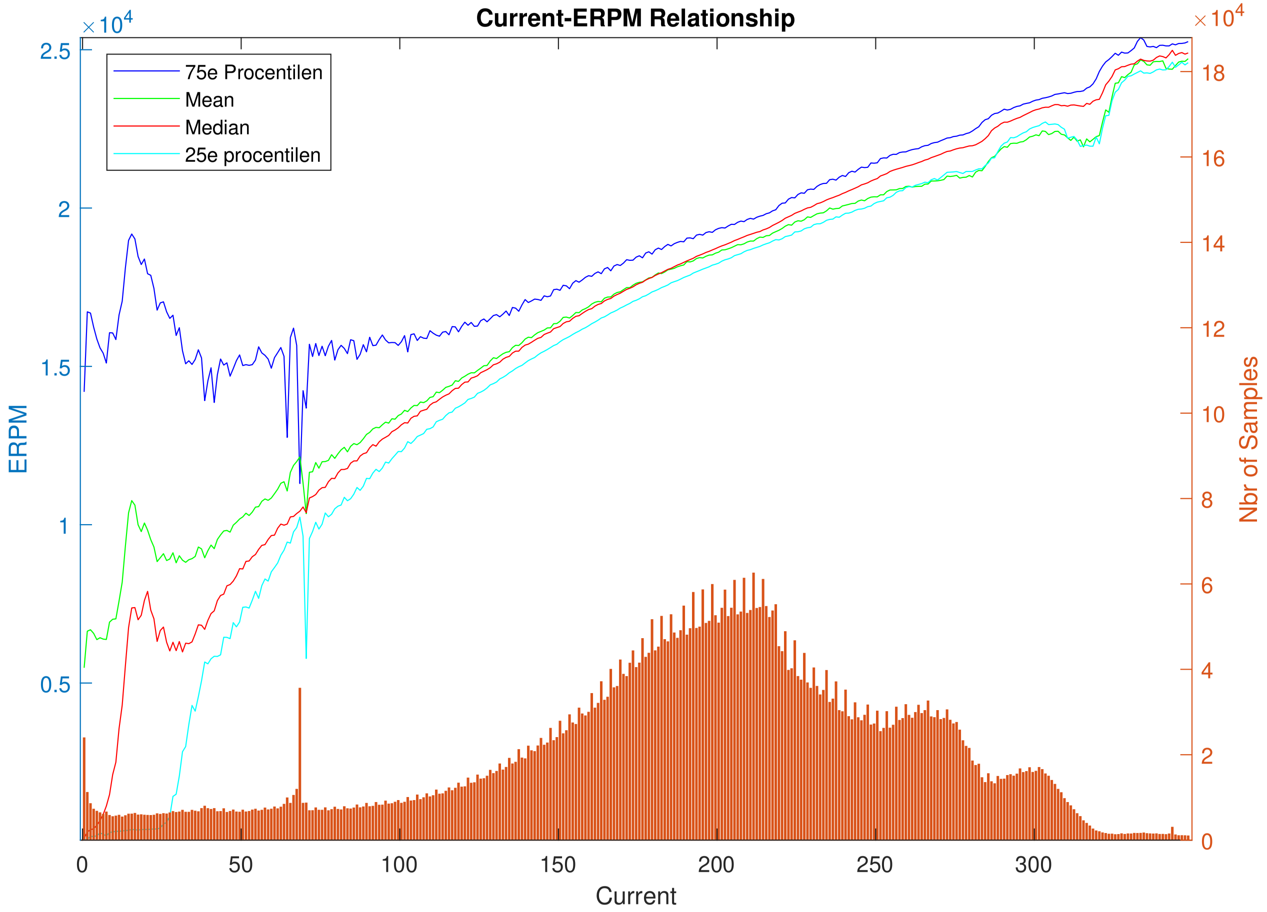

Figure C.1: Visualization of the distribution of eRPM for each possible level of current. This plot illustrates the relationship between Current and ERPM, showcasing various performance metrics, such as the 75th percentile, Mean, Median, and 25th percentile values. The orange bar diagram, plotted against the right axis, shows how many samples there were for each level of current.

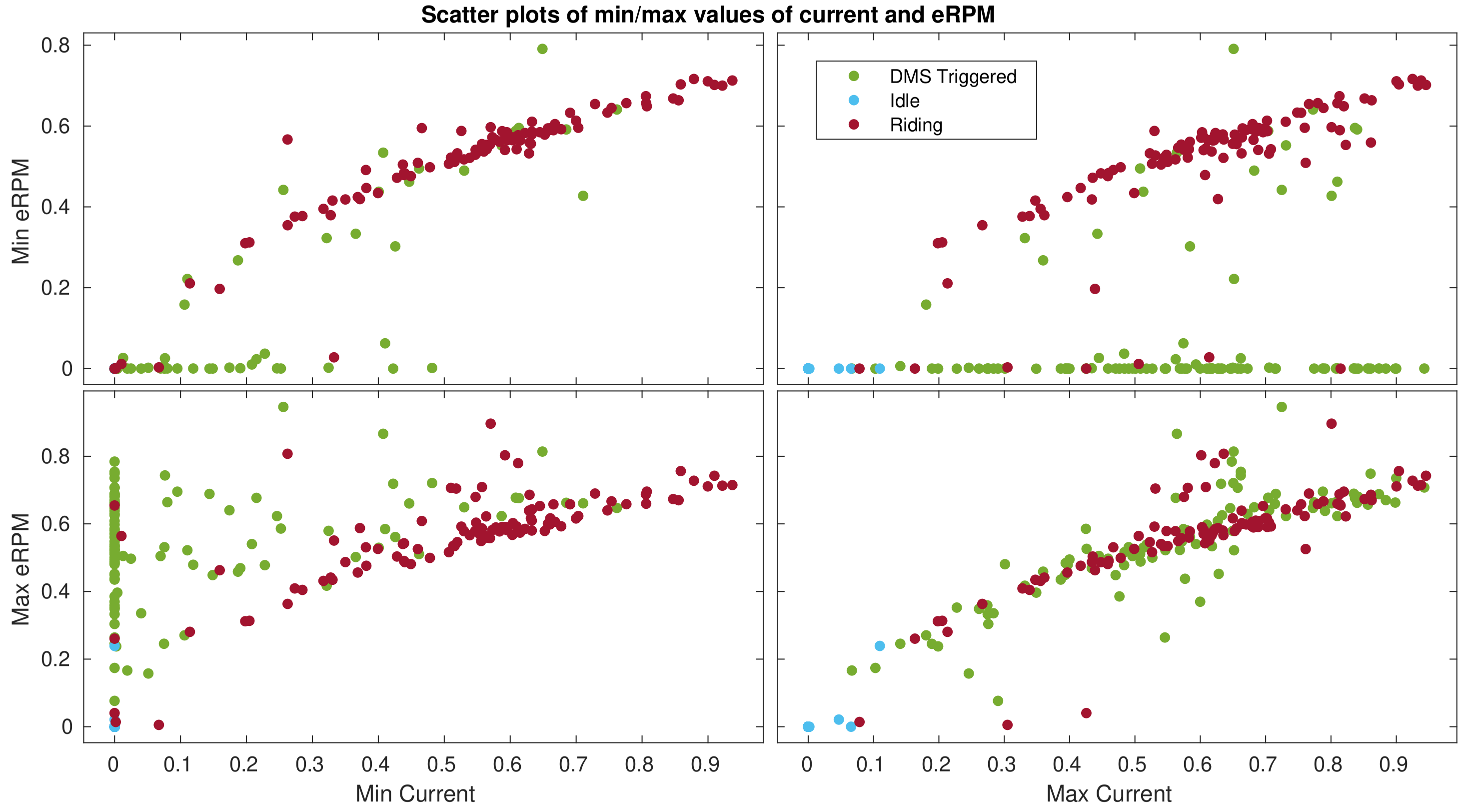

Figure C.2: Scatter plots of the minimum and maximum current and eRPM values.

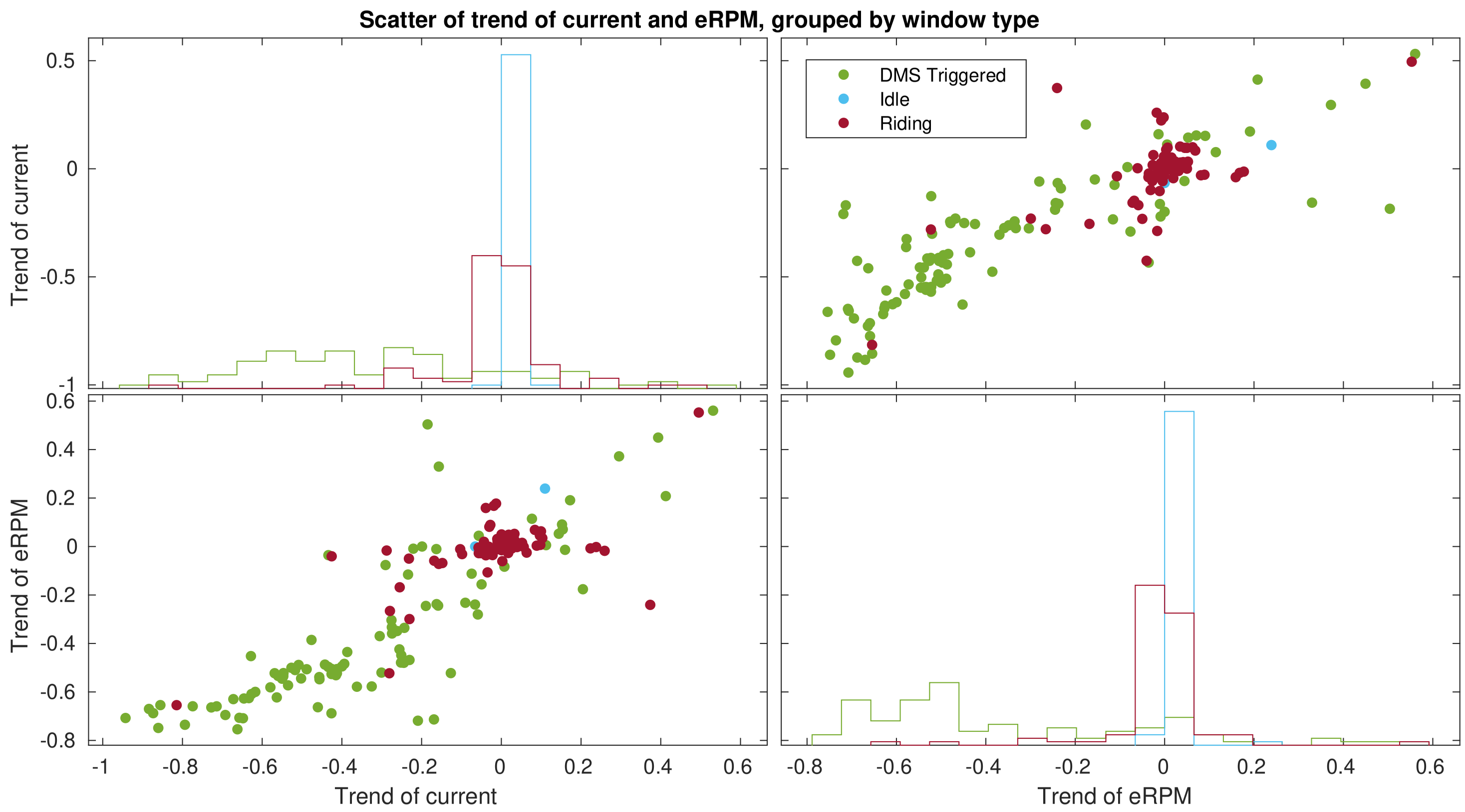

Figure C.3: Scatter plots of the trends of eRPM and current for idle windows, windows where the DMS has triggered, and windows of regular riding of the board. The plots in the negative diagonal show the distribution of each metric.

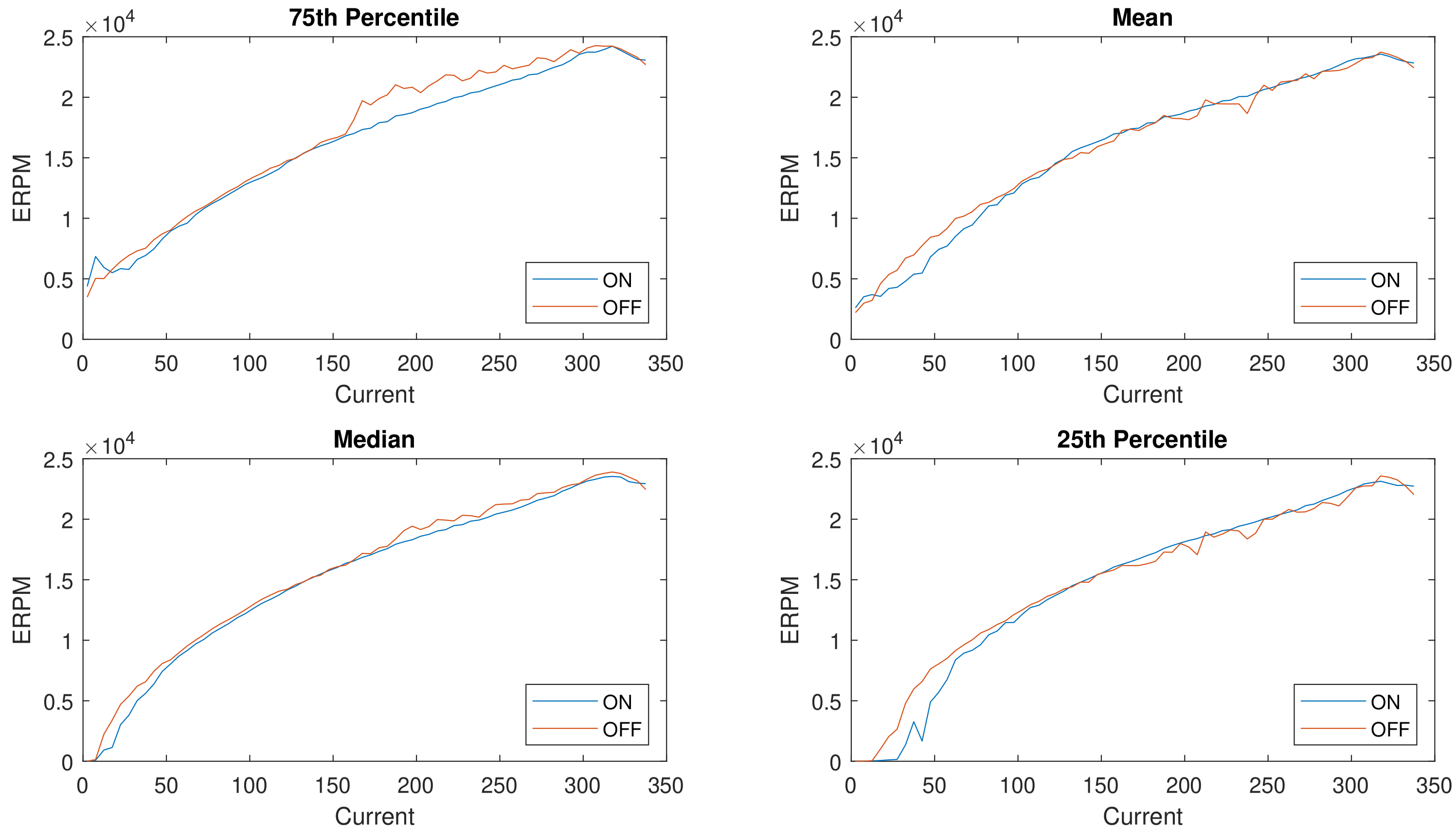

Figure C.4: Visualization ERPM values for each current value with and without rider: The four plot illustrates the relationship between Current and ERPM, each subplot show various performance metrics, such as the 75th percentile, Mean, Median, and 25th percentile values.

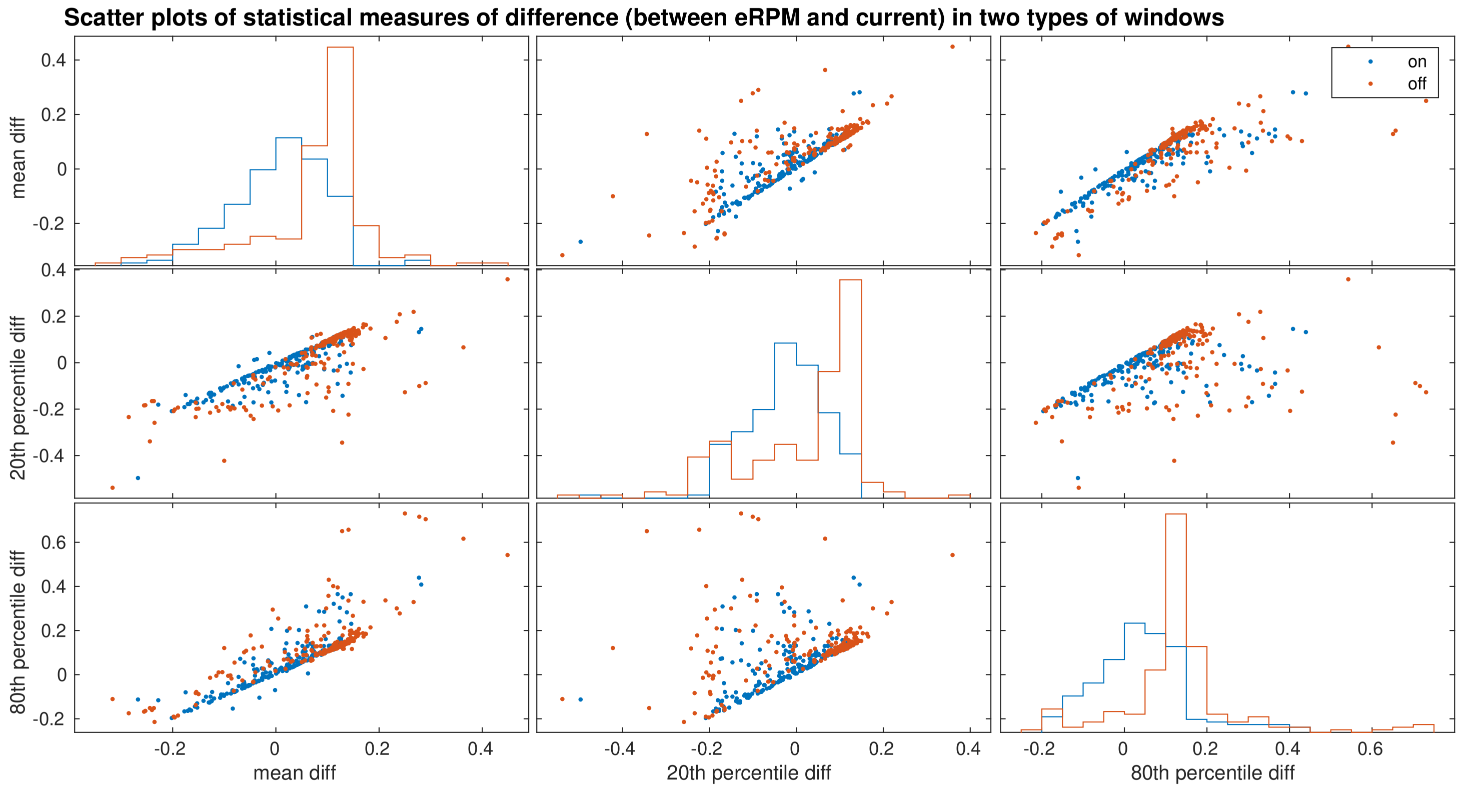

Figure C.5: Scatter plot of mean, 20th percentile and 80th percentile of the difference between eRPM and current for windows of size 10. Grouped by category of window, on-board or off-board. The negative diagonals show the distributions for each feature.

Appendix D

Results for on-board detection

Table D.1: Mean values with 95% confidence intervals for Window 10, Overlap 5.

	PrecisionOn	RecallOn	PrecisionOff	RecallOff
Bagged Trees	**0.90883** [0.90609 - 0.91158]	**0.75782** [0.75518 - 0.76046]	**0.73883** [0.73653 - 0.74114]	**0.9183** [0.91629 - 0.92031]
CNN	0.89533 [0.87996 - 0.91069]	0.74186 [0.72266 - 0.76107]	0.71773 [0.7088 - 0.72666]	0.89245 [0.87314 - 0.91175]
LSTM	0.86099 [0.84084 - 0.88114]	0.72444 [0.68728 - 0.76159]	0.72922 [0.7094 - 0.74903]	0.85353 [0.82328 - 0.88378]
CNN+LSTM	0.89993 [0.88584 - 0.91401]	0.73056 [0.71365 - 0.74747]	0.71131 [0.70281 - 0.71981]	0.90009 [0.88382 - 0.91635]
Fine Tree	0.90865 [0.90444 - 0.91287]	0.73529 [0.73008 - 0.7405]	0.71735 [0.71372 - 0.72097]	0.91271 [0.90882 - 0.91661]

Table D.2: Mean values with 95% confidence intervals for Window 10, Overlap 8.

	PrecisionOn	RecallOn	PrecisionOff	RecallOff
Bagged Trees	0.89332 [0.88955 - 0.89709]	**0.77686** [0.77401 - 0.77971]	**0.75327** [0.75029 - 0.75625]	0.90304 [0.90117 - 0.90491]
CNN	0.90737 [0.89264 - 0.92209]	0.72754 [0.70943 - 0.74565]	0.7116 [0.70322 - 0.71998]	0.90769 [0.88965 - 0.92574]
LSTM	0.85876 [0.83292 - 0.88461]	0.72332 [0.69039 - 0.75626]	0.72687 [0.7087 - 0.74504]	0.84839 [0.80386 - 0.89292]
CNN+LSTM	0.90753 [0.89812 - 0.91694]	0.72442 [0.71081 - 0.73802]	0.7096 [0.70236 - 0.71683]	0.91008 [0.89821 - 0.92196]
Fine Tree	**0.91282** [0.90835 - 0.9173]	0.7334 [0.72874 - 0.73806]	0.71662 [0.71391 - 0.71933]	**0.91621** [0.91132 - 0.92109]

Table D.3: Mean values with 95% confidence intervals for Window 20, Overlap 10.

	PrecisionOn	RecallOn	PrecisionOff	RecallOff
Bagged Trees	**0.91298** [0.90957 - 0.9164]	**0.7521** [0.74917 - 0.75504]	**0.75023** [0.74834 - 0.75212]	**0.92601** [0.92345 - 0.92857]
CNN	0.9051 [0.89225 - 0.91795]	0.72149 [0.70432 - 0.73866]	0.70721 [0.69877 - 0.71565]	0.90682 [0.89135 - 0.9223]
LSTM	0.86213 [0.84331 - 0.88096]	0.71036 [0.6559 - 0.76483]	0.72264 [0.69871 - 0.74657]	0.85755 [0.83166 - 0.88344]
CNN+LSTM	0.90472 [0.89463 - 0.9148]	0.72462 [0.70865 - 0.74058]	0.70919 [0.70066 - 0.71773]	0.90684 [0.89444 - 0.91924]
Fine Tree	0.90629 [0.90152 - 0.91106]	0.72272 [0.7147 - 0.73075]	0.72459 [0.71963 - 0.72955]	0.91613 [0.91067 - 0.92158]

Table D.4: Mean values with 95% confidence intervals for Window 20, Overlap 15.

	PrecisionOn	RecallOn	PrecisionOff	RecallOff
Bagged Trees	0.90033 [0.8968 - 0.90386]	**0.77386** [0.76957 - 0.77815]	**0.76633** [0.76305 - 0.76961]	0.9124 [0.91072 - 0.91407]
CNN	**0.92005** [0.90761 - 0.93248]	0.70158 [0.68422 - 0.71894]	0.69837 [0.68995 - 0.7068]	**0.9246** [0.9102 - 0.939]
LSTM	0.87682 [0.85637 - 0.89727]	0.7164 [0.68513 - 0.74766]	0.7266 [0.71029 - 0.74291]	0.87398 [0.84559 - 0.90237]
CNN+LSTM	0.90305 [0.89376 - 0.91235]	0.72283 [0.70633 - 0.73933]	0.70771 [0.69889 - 0.71652]	0.90541 [0.89375 - 0.91706]
Fine Tree	0.91579 [0.91216 - 0.91942]	0.72787 [0.72173 - 0.73401]	0.73013 [0.72624 - 0.73401]	0.92356 [0.91967 - 0.92746]

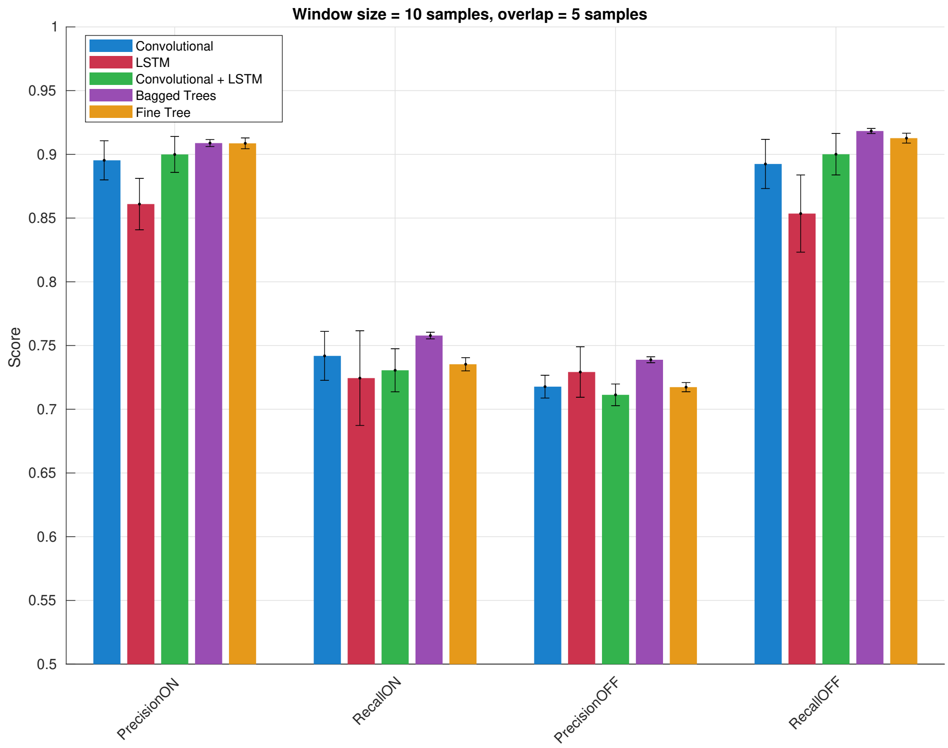

Figure D.1: Mean recall and precision scores, along with 95% confidence intervals, for *riding* and *fall-off* classes for 10-sample windows with 5-sample overlap.

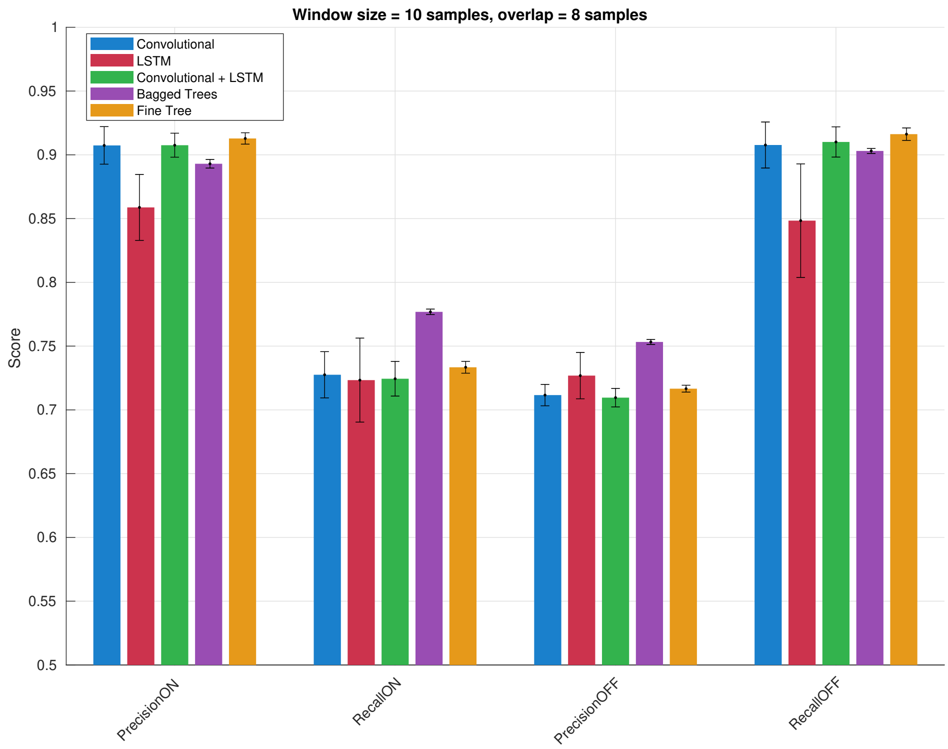

Figure D.2: Mean recall and precision scores, along with 95% confidence intervals, for *riding* and *fall-off* classes for 10-sample windows with 8-sample overlap.

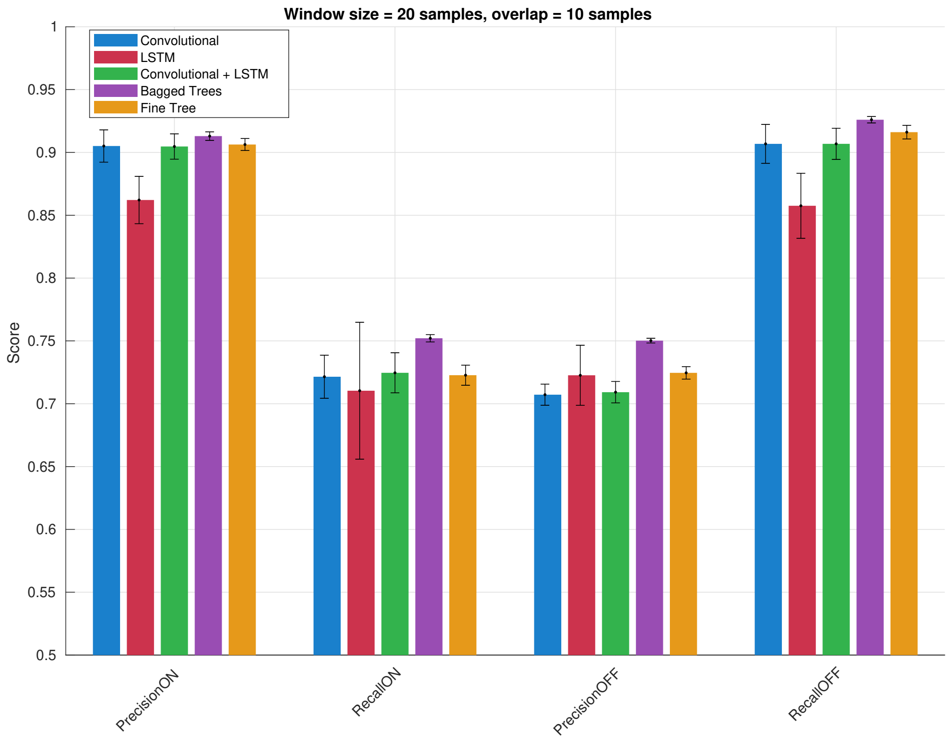

Figure D.3: Mean recall and precision scores, along with 95% confidence intervals, for *riding* and *fall-off* classes for 20-sample windows with 10-sample overlap.

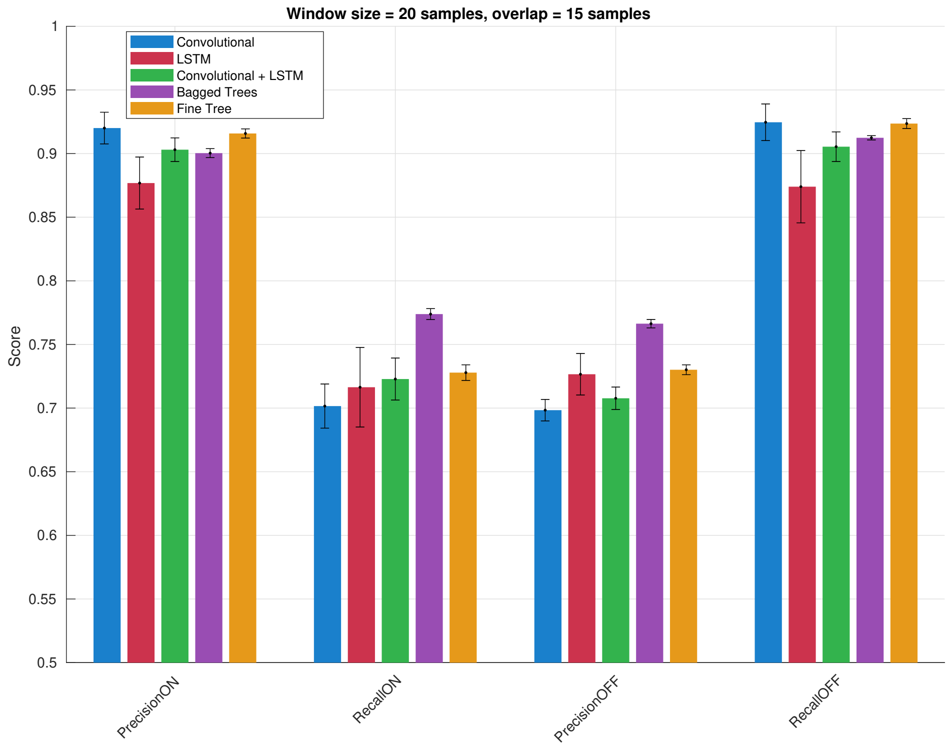

Figure D.4: Mean recall and precision scores, along with 95% confidence intervals, for *riding* and *fall-off* classes for 20-sample windows with 15-sample overlap.

Figure D.5: Section of a ride log where the red areas mark the time stamps covered by windows that the classifier has seen as off-board.

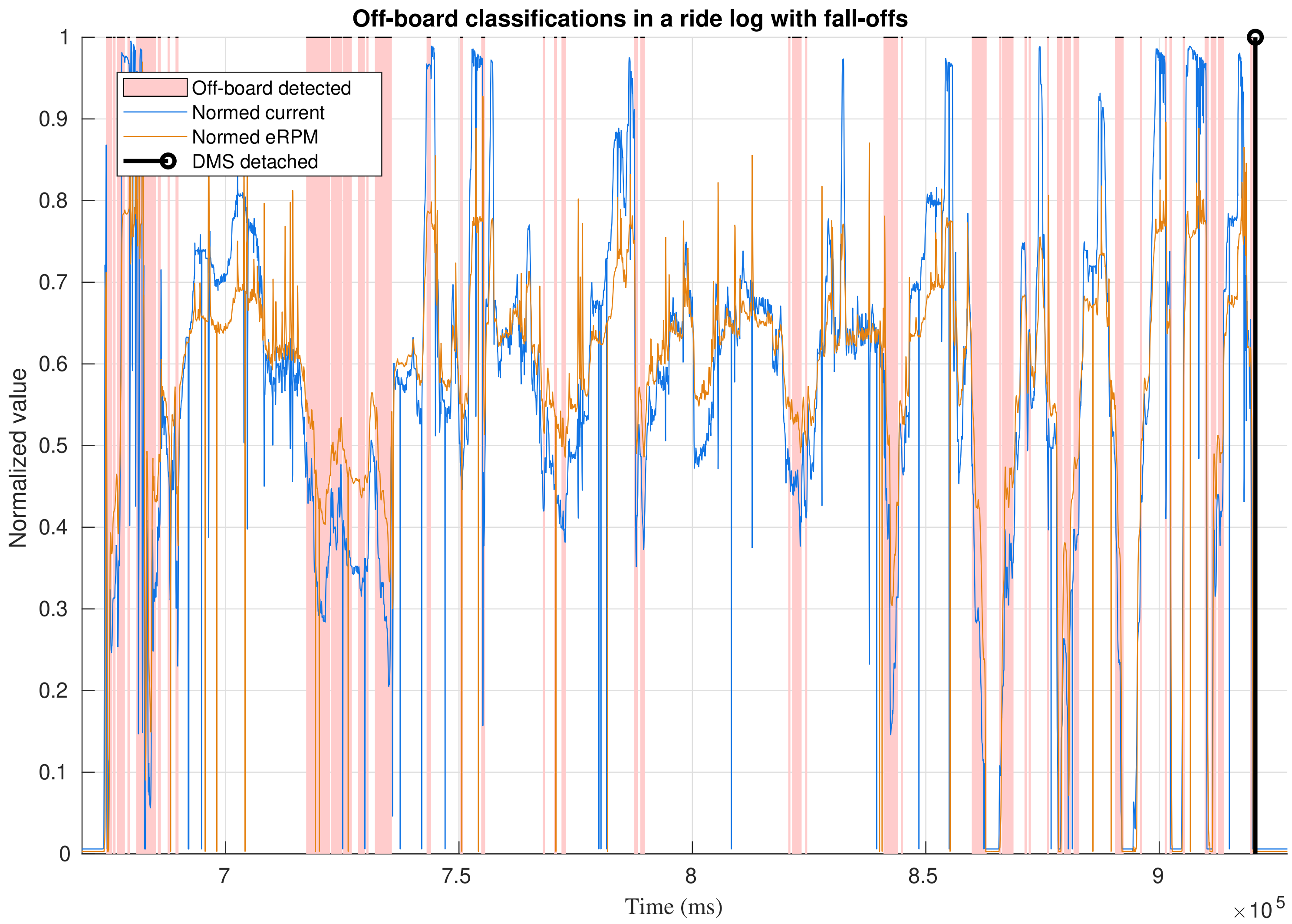

Figure D.6: Another view of the same ride as Figure 4.8. Red areas mark where the classifier has found an off-board window.

Appendix E

Fall-off detection results

Table E.1: Mean values with 95% confidence intervals for Window 2, Overlap 1.

	Precision Riding	Recall Riding	Precision Fall-off	Recall Fall-off
BaggedTree	**0.7140** [0.7067-0.7214]	0.7478 [0.7396-0.7561]	0.7569 [0.7503-0.7635]	**0.7089** [0.7001-0.7180]
CNN	0.6984 [0.6842-0.7126]	**0.7980** [0.7764-0.8197]	0.7772 [0.7628-0.7916]	0.6165 [0.5890-0.6450]
LSTM	0.6806 [0.6610-0.7002]	0.7762 [0.7207-0.8317]	0.7617 [0.7271-0.7962]	0.5880 [0.5370-0.6390]
CNN+LSTM	0.7133 [0.6998-0.7268]	0.7831 [0.7601-0.8061]	**0.7794** [0.7597-0.7991]	0.6430 [0.6170-0.6689]
FineTree	0.7039 [0.6948-0.7131]	0.7643 [0.7458-0.7827]	0.7637 [0.7501-0.7772]	0.6760 [0.6600-0.6927]

Table E.2: Mean values with 95% confidence intervals for Window 3, Overlap 2.

	Precision Riding	Recall Riding	Precision Fall-off	Recall Fall-off
BaggedTree	**0.7145** [0.7048 - 0.7242]	0.7256 [0.7140 - 0.7371]	0.7584 [0.7513 - 0.7655]	**0.7231** [0.7135 - 0.7330]
CNN	0.7079 [0.6838 - 0.7321]	0.7235 [0.6529 - 0.7942]	0.7653 [0.7307 - 0.7999]	0.6610 [0.6140 - 0.7080]
LSTM	0.6717 [0.6576 - 0.6858]	**0.7620** [0.7290 - 0.7950]	0.7579 [0.7379 - 0.7779]	0.6070 [0.5710 - 0.6440]
CNN+LSTM	0.7092 [0.6898 - 0.7286]	0.7343 [0.7036 - 0.7650]	**0.7714** [0.7462 - 0.7967]	0.6800 [0.6450 - 0.7162]
FineTree	0.7003 [0.6916 - 0.7090]	0.7395 [0.7203 - 0.7587]	0.7606 [0.7486 - 0.7727]	0.6880 [0.6700 - 0.7060]

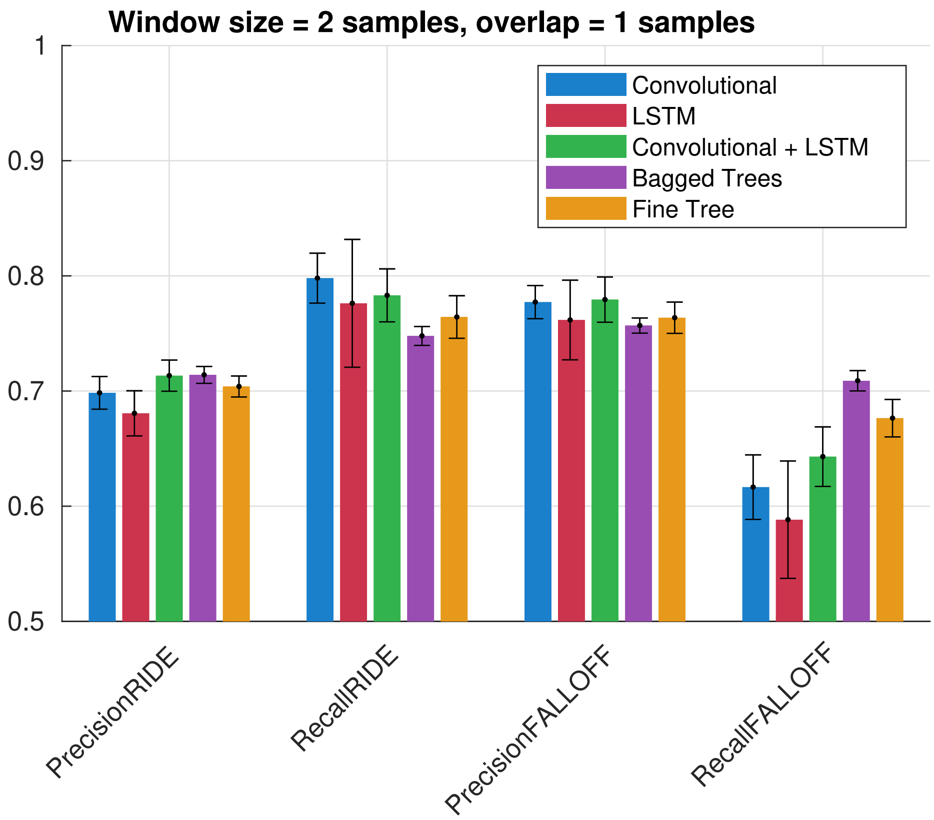

Figure E.1: Mean recall and precision scores, along with 95% confidence intervals, for *riding* and *fall-off* classes for 2-sample windows with 1-sample overlap.

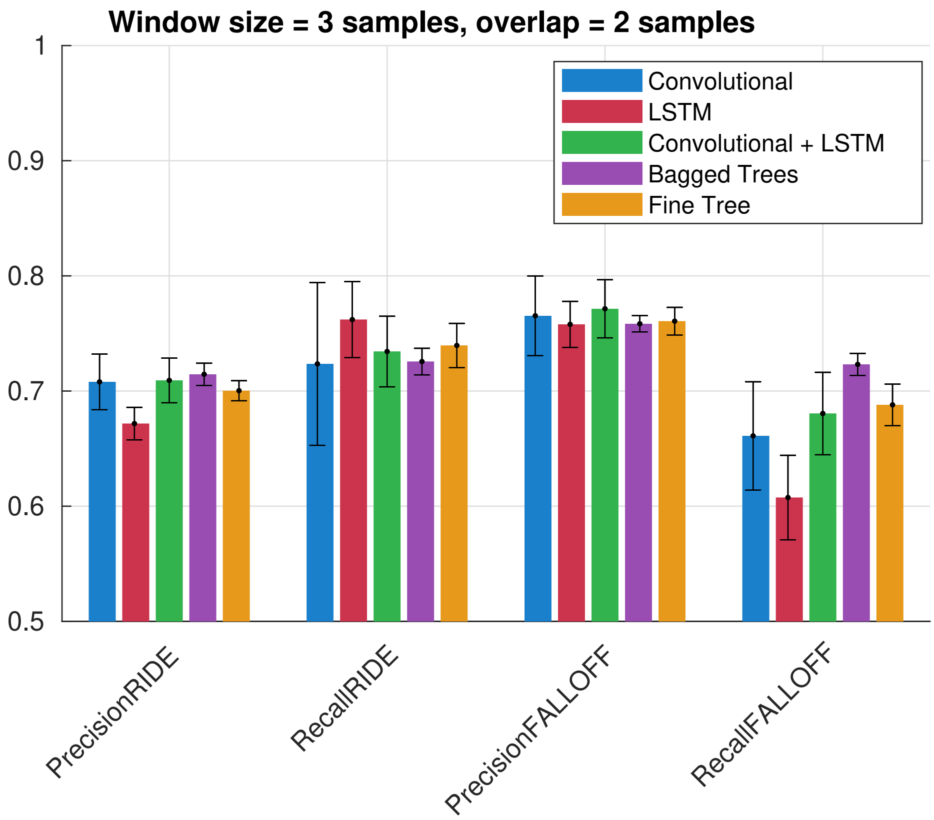

Figure E.2: Mean recall and precision scores, along with 95% confidence intervals, for *riding* and *fall-off* classes for 3-sample windows with 2-sample overlap.

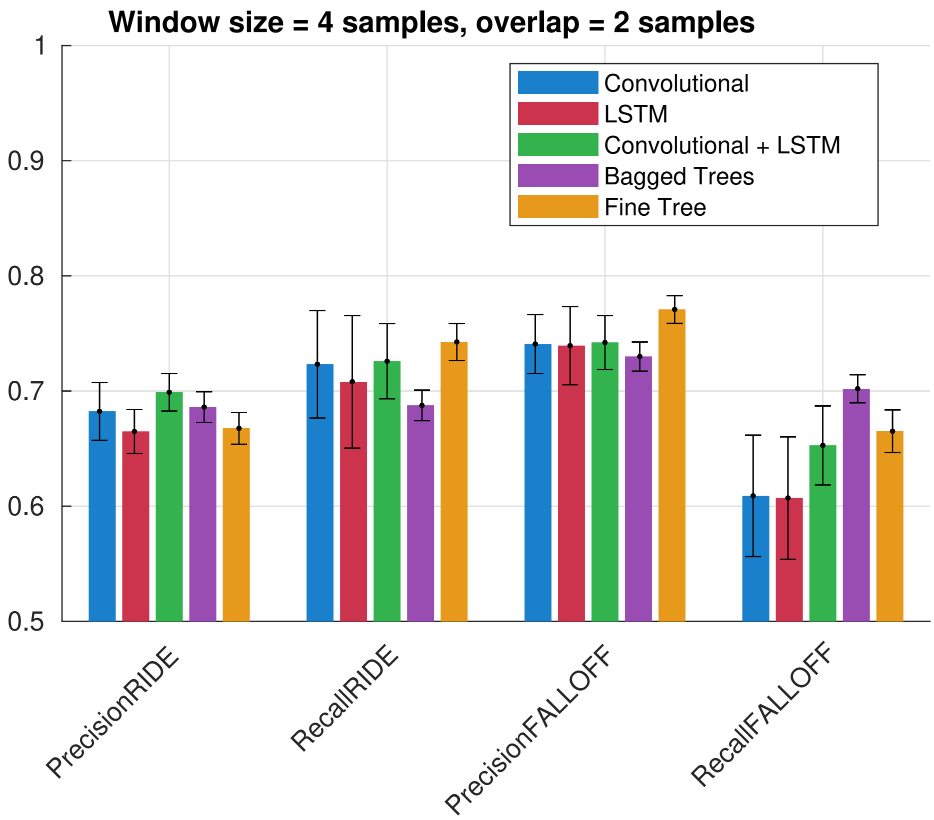

Figure E.3: Mean recall and precision scores, along with 95% confidence intervals, for *riding* and *fall-off* classes for 4-sample windows with 2-sample overlap.

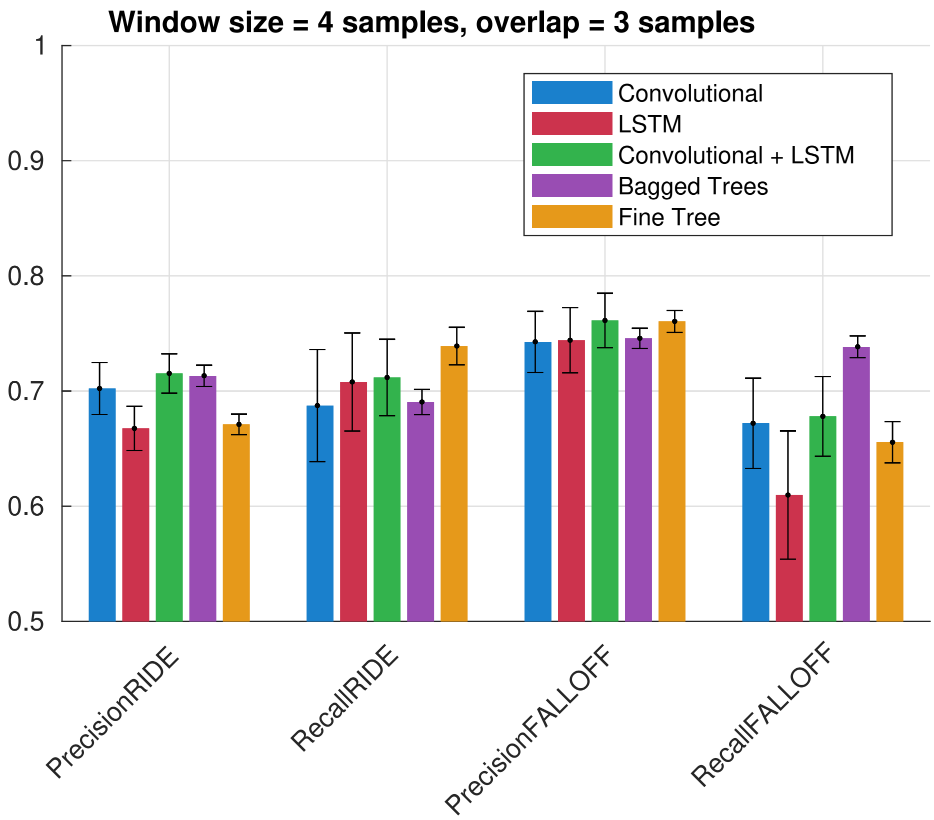

Figure E.4: Mean recall and precision scores, along with 95% confidence intervals, for *riding* and *fall-off* classes for 4-sample windows with 3-sample overlap.

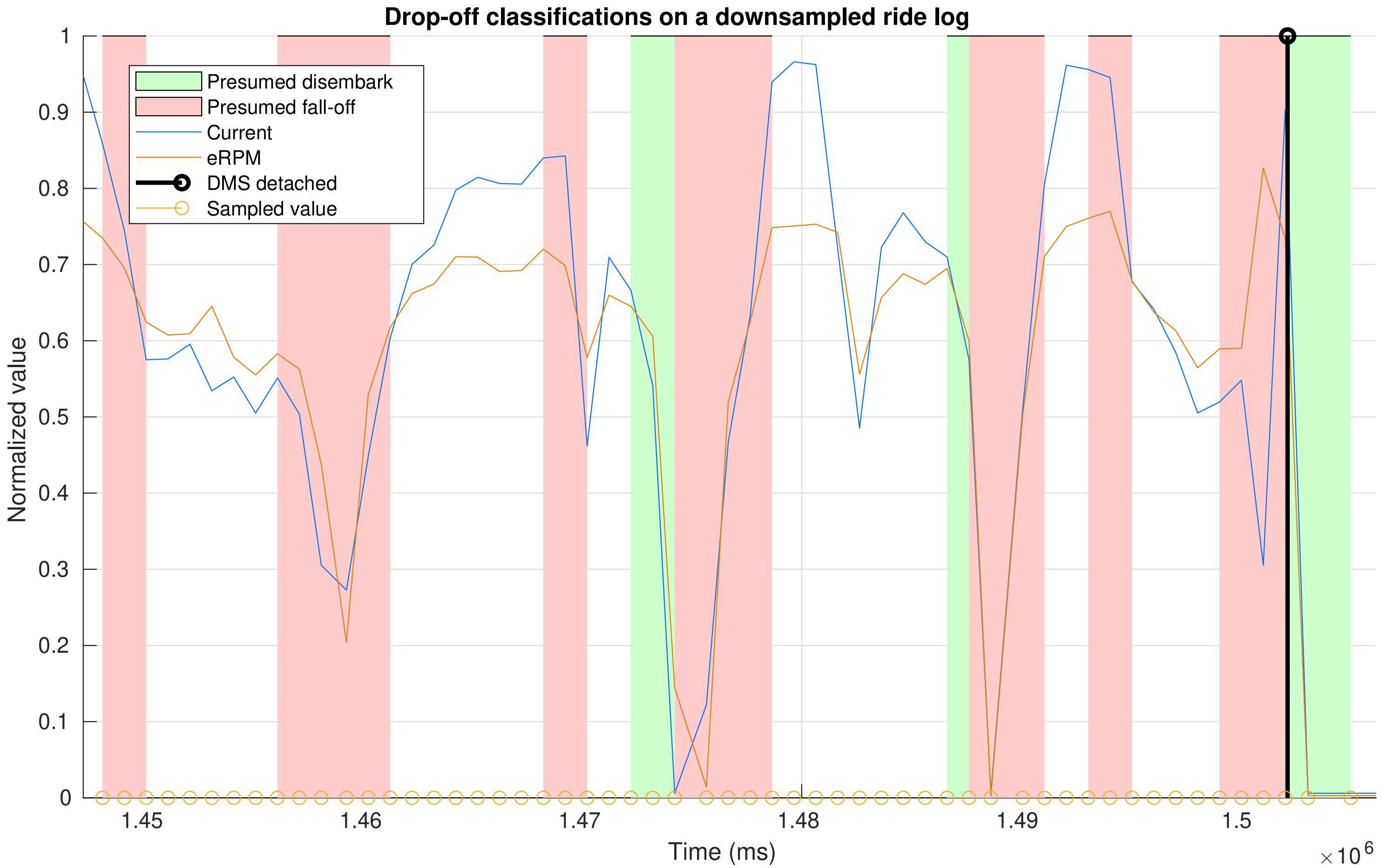

Figure E.5: Visualization of disembark and fall-off classifications on a ride log, down sampled from 20 Hz to 1 Hz, split into 3-sample windows. The light red and light green areas show where the bagged tree classifier found a drop-off. The black vertical bar shows when the DMS was actually unplugged, meaning there was a fall off the board.

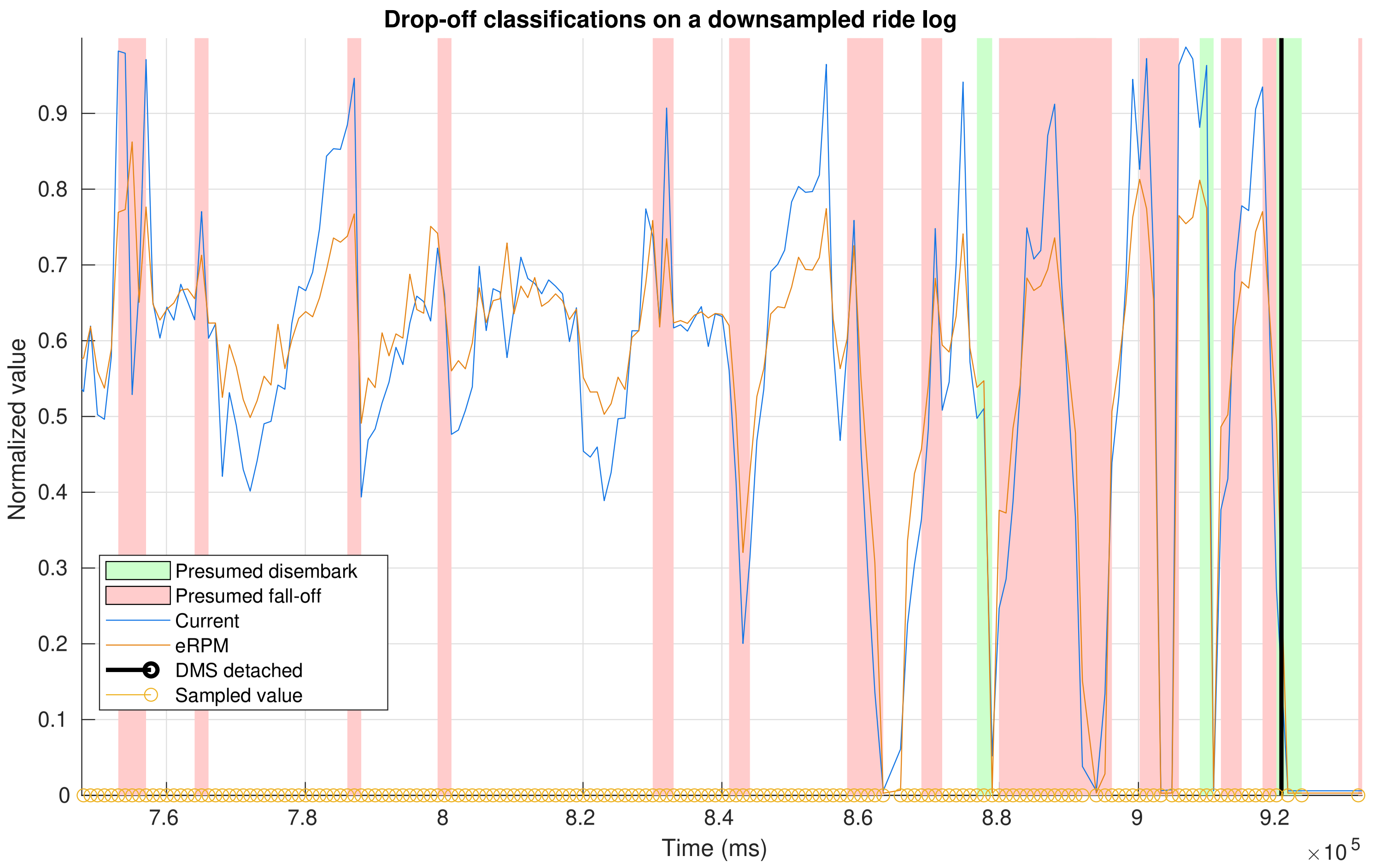

Figure E.6: Another view of the same ride as Figure **??**. Red areas mark windows classified as *presumed fall-offs*, green areas mark *presumed disembarks*. The green circles on the x-axis show the sample times.

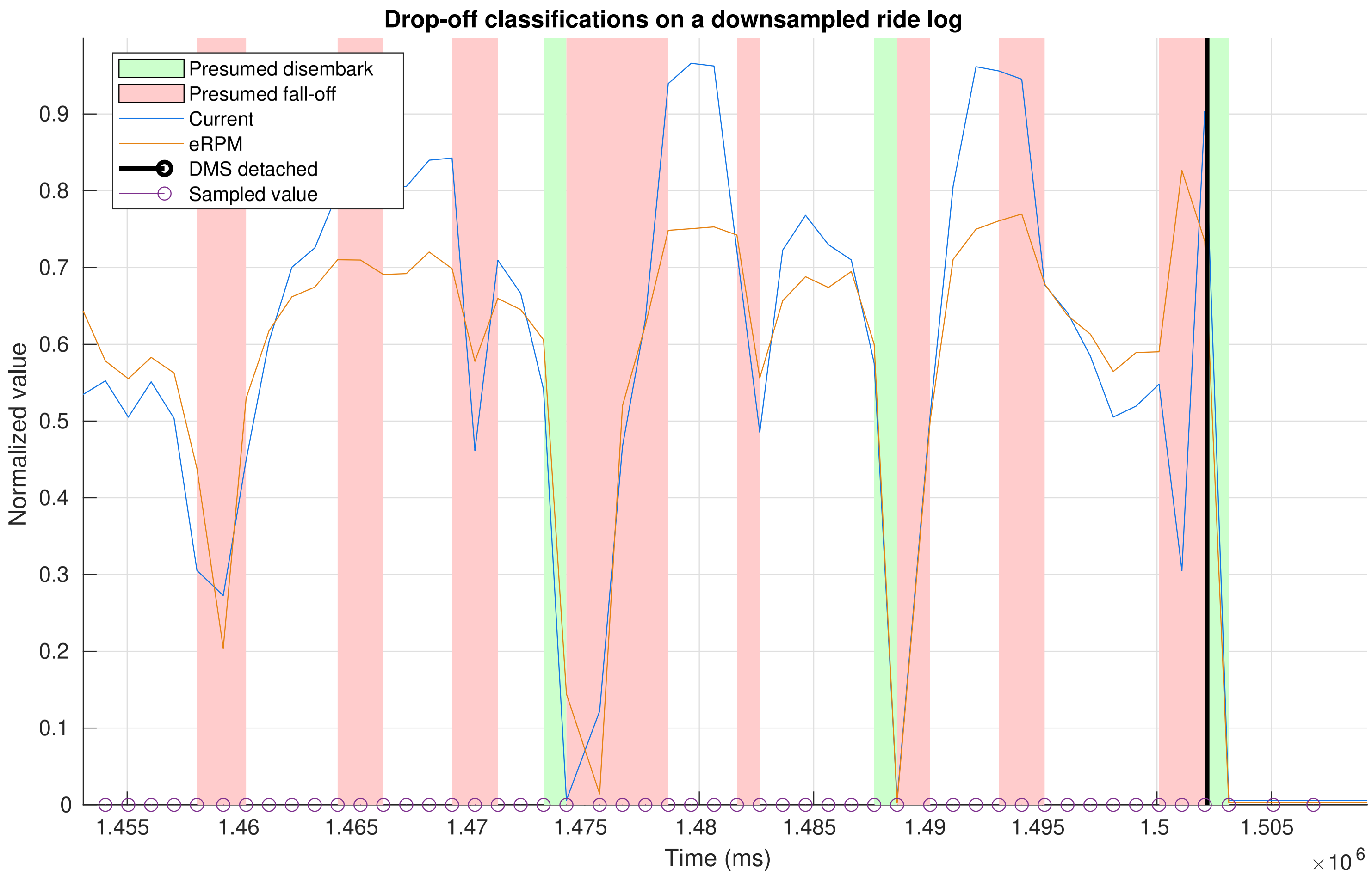

Figure E.7: Visualization of disembark and fall-off classifications on a ride log, down sampled from 20 Hz to 1 Hz, split into 2-sample windows. The light red and light green areas show where the bagged tree classifier found a drop-off. The black vertical bar shows when the DMS was actually unplugged, meaning there was a fall off the board.

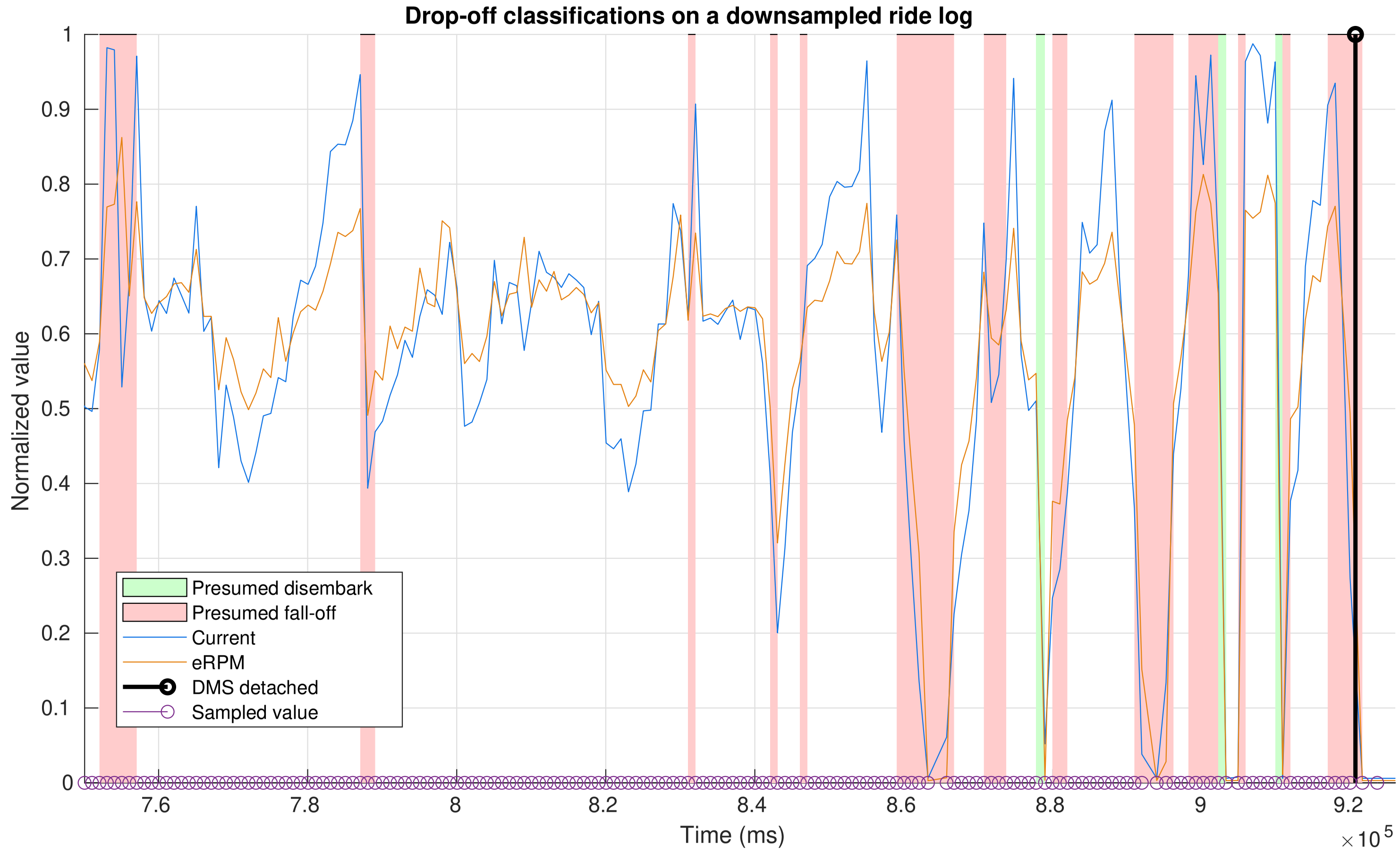

Figure E.8: Another view of the same ride as Figure **??**. Red areas mark windows classified as *presumed fall-offs*, green areas mark *presumed disembarks*. The green circles on the x-axis show the sample times.