
# Real-time Head Movement Tracking through Earables in Moving Vehicles

University of Oulu
Information Processing Science
Master's Thesis
Leyla Shojaeifard
June 2023

# Abstract

The Internet of Things is enabling innovations in the automotive industry by expanding the capabilities of vehicles by connecting them with the cloud. One important application domain is traffic safety, which can benefit from monitoring the driver's condition to see if they are capable of safely handling the vehicle. By detecting drowsiness, inattentiveness, and distraction of the driver it is possible to react before accidents happen. This thesis explores how accelerometer and gyroscope data collected using earables can be used to classify the orientation of the driver's head in a moving vehicle. It is found that machine learning algorithms such as Random Forest and K-Nearest Neighbor can be used to reach fairly accurate classifications even without applying any noise reduction to the signal data. Data cleaning and transformation approaches are studied to see how the models could be improved further. This study paves the way for the development of driver monitoring systems capable of reacting to anomalous driving behavior before traffic accidents can happen.

*Keywords*
Earable, Head Movement Tracking, Machine Learning

*Supervisor*
PhD, Assistant Professor Ella Peltonen

# Contents

# Abbreviations

| | |
|---|---|
| BLE | Bluetooth Low Energy |
| CHAR | Composite Head-body Activities Recognition |
| FHP | Forward Head Posture |
| GPS | Global Positioning System |
| HCI | Human-Computer Interaction |
| IMU | Inertial Measurement Unit |
| IoT | Internet of Things |
| kNN | K-Nearest Neighbor |
| LR | Logistic Regression |
| RFC | Random Forest Classifier |

# 1. Introduction

Our world has been transformed by the rapid developments in technology, and among the most auspicious advancements is the Internet of Things (IoT). It has seamlessly connected physical objects with the digital world, bringing about plenty of opportunities for innovation and improvement for a wide variety of industries. One industry that has seen a lot of growth is the smart device industry (Choudhury, 2021). Devices such as fitness trackers, smart rings and necklaces, smartwatches, earbuds, and other sensor-equipped gadgets have been in demand thanks to their capabilities to monitor various aspects of our lives (Choudhury, 2021). With these smart devices it becomes possible to track and analyse people's movement patterns and to provide them worthy insights into their health, behaviour, and activities.

Interest has been growing in tracking and monitoring people's movement using wearable smart devices for use cases such as traffic safety during the past few years. Road safety could be improved by reducing the risk of accidents through detecting drivers' movement patterns from real-time data. It is possible to use the data from wearable smart devices to identify potential risks, deepen the understanding of driver behavior, and develop proactive measures for mitigating the risks associated with dangerous patterns.

One promising wearable, capable of capturing head orientation data for analysis, is the set of eSense earbuds. The earbuds have a microphone, an accelerometer, a gyroscope, and a Bluetooth Low Energy (BLE) module capable of bidirectional communication (Kawsar, Min, Mathur, Montanari, Acer, et al., 2018; Hossain et al., 2019). These ear-worn wearables, i.e., earables allow for collecting the driver's head movement data comfortably, conveniently, and unobtrusively. The collected data can then be used to classify the head orientation of the driver into left, straight, and right in real time even when the vehicle is moving.

The concept of tracking a person's movement using wearables revolves around the idea of collecting and analyzing data related to their physical activities and behavior. Wearable smart devices are equipped with various sensors, such as ones capable of detecting motion and vibration which could be used for activity recognition, speech and breathing patterns, or possibly facial expressions, and optical sensors that allow monitoring heart rates and blood oxygenation (Powar & Beresford, 2019). There are also accelerometers and gyroscopes, which can precisely measure parameters like speed, acceleration, and orientation. These sensors generate a continuous stream of data that can be analyzed to extract valuable insights into a person's movement patterns. By combining movement data with location information obtained through GPS technology, a comprehensive picture of the person's behavior while driving a vehicle can be obtained.

Traffic and driver safety form a paramount concern in today's fast-paced and ever-evolving transportation landscape, with a staggering number of accidents occurring on our roads each year and the increasing number of vehicles on the roads. The ability to accurately monitor a car driver's condition can play a crucial role in reducing accidents and improving overall road safety. Wearable smart devices provide a unique opportunity to assess a driver's condition in real-time. Factors such as fatigue, drowsiness, distraction (Kawsar, Min, Mathur, Montanari, Amft, et al., 2018), and aggressive driving behaviors can be monitored by analyzing the data collected from these devices. By detecting early signs of driver impairment or risky behaviors, appropriate interventions or alerts can be provided to prevent accidents and promote safer driving habits.

However, the successful implementation of wearable-based driver condition monitoring systems relies heavily on the development of accurate and robust algorithms for analyzing the vast amounts of movement data collected. This is where machine learning pattern recognition techniques come into play. Machine learning algorithms have shown remarkable success in handling complex data and extracting meaningful patterns. By training these algorithms on labeled datasets comprising various driving conditions, it becomes possible to create models that can accurately classify different driver behaviors and conditions, and further to detect anomalies. Then it is only a matter of developing responses for detected anomalies or concerns.

This study is motivated by the need to overcome the limitations of the currently available solutions, and to develop a more robust and non-intrusive solution for real-time head movement tracking in moving vehicles. I aim to utilize eSense earables to accurately detect the driver's head position. By successfully classifying head orientation using eSense earables, the proposed system can facilitate the development of intelligent driver assistance systems capable of detecting problems with the driver's health, capability to drive, or inattentiveness of the traffic in real time. The non-intrusive nature of eSense earables ensures that the driver's privacy is preserved, and the real-time capabilities allow for immediate responsiveness to potential safety risks. According to Choudhury (2021), society has already adopted earables well into people's everyday lives, and they are not as surprising and uncertain technologies as, for example, smart glasses bringing about visual augmented reality.

The outcomes of this research have wide-ranging practical implications across various domains, including the automotive industry, transportation services, and driver training programs. The accurate tracking of head movements can enhance the effectiveness of existing driver assistance technologies, paving the way for safer and more reliable autonomous vehicles. Moreover, the insights gained from this study can be leveraged to develop personalized coaching systems for drivers, thereby promoting safer driving practices. This study also seeks to contribute to the research on earables, which is still in early stages as has been noted by Powar & Beresford (2019).

In this thesis work, I aim to investigate the feasibility and effectiveness of utilizing wearable smart devices and machine learning pattern recognition for monitoring a car driver's condition and enhancing traffic safety. More specifically, I am seeking to use machine learning on the accelerometer and gyroscope data collected from eSense earbuds to determine the head orientation of the person in a moving vehicle. Future research and development can use the head orientation information to detect patterns related to the driver's well-being and condition and to produce responses to anomalous behaviors.

This study answers the following research question:

> **RQ1** How to effectively classify the head orientation of a car driver into left, straight, and right using data from eSense device?

To answer this research question, several sub-questions (SUB-RQs) have been identified:

> **SUB-RQ1.1** What machine learning algorithm would be suitable for addressing this classification problem?
>
> **SUB-RQ1.2** How to clean the data before utilizing it for machine learning purposes?

**SUB-RQ1.3** How to transform the data to improve the prediction capabilities of the models?

This thesis is structured as follows: chapter 2 provides an overview of the existing literature related to earables and head movement tracking through Inertial Measurement Unit (IMU) sensors, along with driver monitoring systems. Chapter 3 presents the research approach adopted in this thesis. Chapter 4 presents the findings and the analysis conducted, alongside responses to the research questions. Chapter 5 discusses the entire study, challenges, contributions, limitations. Lastly, chapter 6 draws the conclusion for the thesis.

# 2. Related Work

In the scope of this thesis, my particular focus lies on identifying head direction using sensor data obtained from accelerometer and gyroscope embedded in eSense device. Therefore, the related work chapter primarily centers on exploring earables and the landscape of their applications. In this section, I also delve into an exploration of various research endeavors focused on tracking head movements for diverse purposes, employing different methodologies. Lastly, I review the existing literature that focuses on driver sensing and explores various tools and approaches employed to identify hazardous driving behaviors.

My objective is to gain insights into the array of head movement tracking and driver monitoring techniques and identify the gap in the literature. Furthermore, I aim to uncover the challenges encountered during head movement tracking and driver monitoring, along with the corresponding strategies employed to address them. By understanding these challenges and solutions, I aspire to proactively mitigate their impact or reduce their potential consequences in my work. Through this comprehensive investigation, I seek to establish a foundation for the successful implementation of my own head movement tracking and driver monitoring system while leveraging the valuable lessons and experiences derived from prior research efforts.

## 2.1 Earables

The age of earables has emerged (Min, Mathur, et al., 2018a). Earables, ear-worn wearable devices (Powar & Beresford, 2019), are a relatively recent concept and have significant untapped potential in various research areas (Ferlini et al., 2019). Through their placement within the ear, earables have the capability to track not only body movement but also specific motions of the head and mouth (Kawsar, Min, Mathur, Montanari, Acer, et al., 2018).

Röddiger et al. (2022) have conducted a systematic literature review on earables and have devised a taxonomy of different phenomena that can be detected within, on, or in the vicinity of the ear. Carrying out in-depth analysis, Röddiger et al. (2022) discovered 13 primary phenomena that serve as the basis for all other related phenomena and discuss diverse sensors and sensing principles employed to identify these phenomena. Röddiger et al. (2022) then categorize the phenomena into four main areas, namely (i) physiological monitoring and health, (ii) movement and activity, (iii) interaction, and (iv) authentication and identification. This categorization greatly influenced me while constructing this particular section. I made an effort to classify each application of earables and their corresponding reviewed articles into their respective categories.

Drawing inspiration once again from Röddiger et al. (2022), Figure 1 presents a summary of the main domains, illustrating various phenomena and the corresponding sensors employed for their capture.

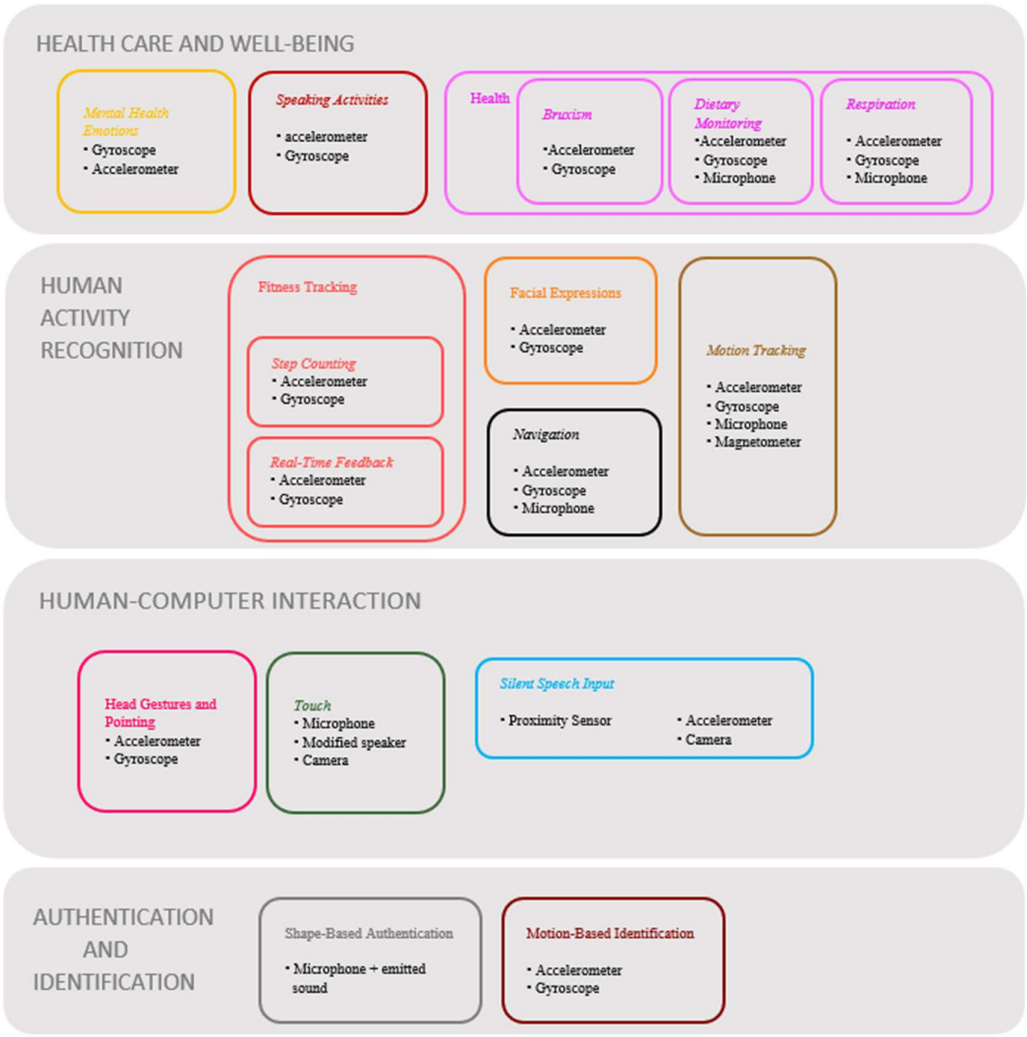


**Figure 1.** Earable sensing overview.

In this section, I individually explore the domains presented in Figure 1, thoroughly examining studies that I have identified as belonging to each domain. I also provide useful insights and details regarding each study.

### 2.1.1 Health Care and Well-Being

Earables have gained significant popularity in the healthcare and well-being field, being extensively utilized to address three key dimensions of well-being: physical, mental, and social well-being. (Kawsar, Min, Mathur, Montanari, Acer, et al., 2018).

Researchers have shown considerable interest in dietary monitoring within the realm of physical health, as evidenced by the abundance of literature dedicated to this topic. The initial phase of dietary monitoring involves detecting eating events, which can be accomplished through either an audio-based or motion-based detection approach (Röddiger et al., 2022). In a study conducted by Lotfi et al. (2020), the researchers examined how effectively the eSense device's audio and inertial sensors can detect chewing events, and to compare the accuracy of each sensing modality. They reported that both audio-based and motion-based approaches for detecting eating events are

susceptible to signal noise caused by unrelated body movement (Lotfi et al., 2020). In a similar study, Bi et al. (2018) introduced Auracle, dietary monitoring system based on earable technology that is designed to automatically recognize the specific timing and duration of a person's eating activities. The Auracle seems to be an audio-based eating detection system, as it employs a contact microphone positioned behind the ear to capture chewing sounds, which are subsequently processed by a customized analog/digital circuit board (Bi et al., 2018). Simultaneously, Min, Mathur, et al., (2018b) demonstrated an audio-kinetic model that utilizes earable devices to monitor the dietary intake of users. The model uses acoustic data, gained from earables, to detect activities related to the intake of food, chewing, and drinking (Min, Mathur, et al., 2018b).

In addition to their application in dietary monitoring, earables are also utilized in various other facets of physical well-being. By incorporating an accelerometer and gyroscope, which enables them of capturing head and neck movement, earables can diagnose Forward Head Posture and provide corrective posture feedback when identifying instances (Radhakrishnan et al., 2021). Furthermore, earables can be used to monitor respiratory rates using their built-in IMUs (Roddiger et al., 2019). Moreover, due to the correlation between jaw movement and deformations in the shape of the ear canal, the use of IMU sensors within the ear canal can enable the detection of jaw clenching and teeth grinding (Roddiger et al., 2019).

Researchers seem to prioritize physical well-being over mental and social well-being, resulting in a lack of comprehensive studies dedicated to these domains. Regarding social well-being, Min, Montanari, et al. (2018) introduced a cross-modal approach that combines audio and physiological data obtained from multi-sensory earables to "detect speaking activities, stress and emotion, and participants in the conversation". The system consists of three models, BLE, motion, and audio. The BLE model identifies a potential gathering of individuals engaged in conversation and determines the participants involved. The motion model detects speaking activities, i.e., determining whether a user is engaged in speech or not. The audio model identifies the possibility of a conversation taking place (Min, Montanari, et al., 2018). A similar study, but in the context of mental well-being, was carried out by Purabi et al. (2019). In their paper, the authors proposed a system that leverages earables and machine learning algorithms to develop a real-time solution for emotion and trait identification using head movement data.

### 2.1.2 Human Activity Recognition

Movement detected around the ear can be organized into separate categories, providing valuable insights into a user's posture, movement patterns, and the specific activities they are involved in (Röddiger et al., 2022). Several studies explore the application of earables as a means of collecting comprehensive data on user movement and their involvement in specific activities.

In a study carried out by Hossain et al. (2019), the researchers used accelerometer sensor built in eSense device to detect different types of activities associated with head and neck motion, including speaking, eating, headshaking, nodding, staying, and walking. They explored data classification using both machine learning (such as Support Vector Machine, Random Forest, and K-Nearest Neighbor algorithms) and deep learning (Convolutional Neural Network) techniques and achieved a high level of accuracy in recognition. Similarly, Min, Mathur, et al. (2018a) achieved 0.80 accuracy in detecting head motion (nodding, shaking) and a 0.95 F1-score in accurately recognizing movement states of a user (stationary, walking, stepping up, stepping down) through the combined

use of accelerometer and gyroscope data, employing a k-Nearest Neighbor classifier. To assess a user's understanding of an online lecture, Kim et al. (2021) employed the accelerometer and gyroscope data from an earable device to classify whether the user was looking at a monitor or down at the desk. Impressively, they achieved F1-scores of approximately 0.92 and 0.90 respectively for these classifications.

The studies mentioned earlier explore the potential of earables in successfully identifying diverse activities (Röddiger et al., 2022). However, the capabilities of earables extend beyond the identification of simple activities. They also have the ability to discern facial expressions. Lee et al. (2019) applied inertial signals captured by an ear-worn device to detect two significant facial expressions: smile and frown. They developed three distinct learning models, namely hidden Markov models, deep neural networks, and shallow models, for automated identification of these facial expressions. Notably, the hidden Markov model achieved an impressive F1-score of 0.85 (Lee et al., 2019). Similarly, Laporte et al. (2021) used an end-to-end deep neural network to classify five verbal and non-verbal activities (nodding, speaking, eating, staying, and head shaking) with an F1-score of 0.82.

In addition to detecting movement through an accelerometer and gyroscope, EarphoneTrack goes a step further by utilizing both wired and wireless earphones for acoustic motion tracking. This innovative approach enables real-time determination of users' movement with an exceptional level of accuracy, measured in millimeters (Cao et al., 2020). The researchers devoted their efforts to tackle various challenges associated with this methodology, such as addressing self-interference in wired earphones, managing frequency offset in wireless earphones, and effectively utilizing the limited bandwidth available for acoustic signals (Cao et al., 2020).

According to Röddiger et al. (2022), navigation can also be accomplished by utilizing the inertial sensors within an earable device to precisely monitor users' position and orientation in space and time, eliminating the need for dependency on a GPS connection. In this regard, Ahuja et al. (2021) introduced PilotEar, "the first end-to-end earable-based inertial navigation system" that effectively gathers motion data from a 9-axis IMU (comprising an accelerometer, gyroscope, and magnetometer) embedded in an earable device, and transmits this data via BLE for real-time monitoring and analysis. The average tracking drift is reported as 0.15 ms for a single earable device and 0.11 ms when combining data from both earables (Ahuja et al., 2021). Prior to their study, Matsumura & Okada (2019) also explored navigation through earables, but with a different focus. The researchers aimed to assess the effectiveness of three-dimensional audio cues delivered through earable devices in guiding blindfolded individuals to change their walking direction. The results demonstrated that the acoustical manipulation successfully minimized deviations and helped maintain a straight walking trajectory in both subtle and overt conditions (Matsumura & Okada, 2019). Notably, there is a significant distinction between the approaches of these two studies: the former being a kinetic system and the latter an acoustic system.

In addition to their role in classifying physical activities, earables have the potential to contribute to the improvement and management of a user's fitness and health (Röddiger et al., 2022). In their study, Prakash et al. (2019) provided evidence of the initial feasibility of step counting at the ear, showcasing accurate tracking across various walking speeds, including very slow, slow, normal, and running, with a remarkable 95% accuracy rate. Furthermore, they proposed a method for detecting and measuring jumping as an indicator of a user's physical health (Prakash et al., 2019). In a similar context, earables also have the capability to offer users feedback during physical activities and

moreover, evaluate their performance afterward (Röddiger et al., 2022). Radhakrishnan & Misra (2019) investigated the utilization of earable devices, commonly used by individuals at gyms, to offer personalized and quantified insights and feedback to users during their workout routines. Their proposed system involves individuals using earables to monitor their activity and physiological context, while the gym equipment is fitted with cost-effective IoT sensors to track the motion dynamics of each piece of equipment (Radhakrishnan & Misra, 2019).

### 2.1.3 Human-Computer Interaction

The earable platform offers a fascinating opportunity to explore innovative and distinctive ways of interaction between humans and computers, thanks to its abundant and varied sensing capabilities (Röddiger et al., 2022). Over the past few years, researchers have shown significant interest in examining the potential of earables to detect various input modes.

One of the most recent works in the field is carried out by Srivastava et al. (2022). In their demo, the researchers presented a technology that employs twin-IMU, positioned behind the ear and on the Temporomandibular Joint, to detect commands that are not verbally expressed. Their proposed system aims to extract jaw motion signals, identify phoneme locations within commands, and reconstruct words using an innovative algorithm. Initial findings demonstrated that the suggested system achieves a word recognition accuracy of ~ 0.95 under noise-free conditions, and ~ 0.93 and ~ 0.91 accuracy during head nodding and walking, respectively. An earlier study by Xu et al. (2020) also explored a novel approach to identifying input by utilizing the integrated components of standard earphones. They introduced EarBuddy, a real-time system that uses the microphone embedded in earphones to detect various tapping and sliding gestures on and around the ear, achieving an accuracy rate of 0.90 by using deep learning classification techniques based on mel spectrograms.

The studies discussed employ input methods that involve either the mouth (mouth-based interaction) or touching areas near the ear (Röddiger et al., 2022). However, according to Röddiger et al. (2022), the motion and direction of the head also offer a hands-free method of input when the hands are occupied or inaccessible.

Odoemelem et al. (2019) used head movements, captured by eSense device, as a means to manipulate a robot arm. Their proposed system undergoes several procedures during its operation. The acceleration data captured by the IMU is converted from the earth's reference frame to the IMU body frame. To address unwanted noise and drift in the accelerometer and gyroscope data, filtering techniques like low-pass and high-pass filters are implemented accordingly. Calibration is conducted to accommodate variations in users' individual head orientations, and mapping is utilized to associate the estimated pitch and roll angles with the corresponding pitch and yaw movements of the robot arm (Odoemelem et al., 2019). In parallel with their study, Katayama, Mathur, Van Den Broeck, et al. (2019) explored the potential of earable devices for enabling more context-aware and user-centric conversational agents through the tracking of sound, movement, and proximity. they introduce a situation-aware conversational agent that dynamically modifies its conversational style, tone, and volume according to the user's emotional state, surrounding environment, social context, and level of activity. This adaptation is achieved by analyzing real-time speech patterns, ambient sounds, and body movements, obtained from eSense device (Katayama, Mathur, Van Den Broeck, et al., 2019; Katayama, Mathur, Okoshi, et al., 2019).

The mentioned studies conducted by Odoemelem et al. (2019) and Katayama, Mathur, Van Den Broeck, et al. (2019) adopt input mechanisms that utilize head gestures and direction, enabling interaction without the need for hands (Röddiger et al., 2022).

### 2.1.4 Authentication and Identification

The classic approach to securing sensitive information on mobile devices revolves around utilizing the user's biometric data, such as fingerprints, for protection (Röddiger et al., 2022). Several studies have explored the potential of earables for authentication and identification purposes.

The study carried out by Clarke et al. (2020) is one of the notable contributions in the means of identification through earables. In their paper, Clarke et al. (2020) introduced a sensor fusion approach that correlates the motion of an earable, measured by its accelerometer, with the head movement of a user in the camera view to provide private audio channels in public settings. In their laboratory experiment, they examined seven distinct movements performed at three different speeds. The results demonstrated an 0.86 accuracy in successfully identifying an individual from a group of 10 participants. Prior to their work, Gao et al. (2019) explored the characteristics of the sound of the ear canal to develop an earable biometric authentication method. They proposed EarEcho authentication system that uses the earpiece speaker and microphone embedded in the earphone to enable acoustic sensing, then it utilizes a two-class Support Vector Machine classifier to perform the authentication task. By implementing a proof-of-concept prototype and conducting tests on 20 participants in different scenarios, the researchers found that EarEcho achieved high recall and precision rates for both one-time and continuous authentication.

Despite both mentioned studies utilizing earables for authentication and identification purposes, there is a major difference in their approaches. The system introduced by Clarke et al. (2020) is motion-based, relying on tracking movements for identification, while EarEcho by Gao et al. (2019) is shape-based, leveraging distinctive ear biometrics for authentication (Röddiger et al., 2022). Therefore, the sensors used in these systems differ. The former employs an accelerometer capable of detecting motion, whereas the latter utilizes a speaker and microphone for acoustic sensing.

## 2.2 Head Movement Tracking

In this section, studies that specifically employ IMU sensor data to track head movement for various purposes are explored.

One of the latest works in the domain is the study conducted by Han et al. (2023). In their paper, the researchers introduce HeadMon, a state-of-the-art system that predicts riding maneuvers to improve safety in micro-mobility, especially for riders wearing helmets. HeadMon includes an IMU integrated into the helmet, allowing it to monitor riders' head motion and predict their following actions (Han et al., 2023). In parallel with their work, Zhu et al. (2023) propose CHAR, a system that uses IMU data to detect composite hea-body movements. However, there is a slight difference in the placement of the IMU between these two studies. The latter study employs an IMU integrated into the eSense device, whereas the former study incorporates an IMU within a helmet. Both studies employ machine learning techniques, but they take different approaches. In CHAR, Zhu

et al. (2023) develop a multi-task learning network with notably high accuracy rates, while in HeadMon, Han et al. (2023) focus on utilizing a deep learning architecture.

Similar to Zhu et al. (2023), Ferlini et al. (2019), Odoemelem et al. (2019), Radhakrishnan et al. (2021), Kim et al. (2020), and Purabi et al. (2019) have all utilized eSense in their respective studies to monitor head movements. However, their purposes and approaches vary significantly. Table 1 presents a comparison of methodologies and applications employed in various studies focusing on head movement tracking.

**Table 1**. Methodologies and applications for head movement tracking in different studies.

| Reference | Device Used | Method | Application |
|---|---|---|---|
| Ferlini et al. (2019) | eSense | complementary filter | detecting visual attention |
| Odoemelem et al. (2019) | eSense | complementary filter | controlling a robot arm |
| Purabi et al. (2019) | eSense | used the Auto-Weka package | identifying physical traits and emotions |
| Kim et al. (2020) | eSense | random forest | recognizing the perceived level of understanding online lectures |
| Radhakrishnan et al. (2021) | eSense | custom algorithm | detecting FHP |
| Han et al. (2023) | IMU within a helmet | HeadMon model (deep learning) | predicting riding maneuvers |
| Zhu et al. (2023) | custom earables | multi-task learning | detecting composite hea-body movements |

Ferlini et al. (2019) and Odoemelem et al. (2019) employ a complementary filter technique to accurately track head movements using data from an eSense earbuds. The left earbud is equipped with an accelerometer and gyroscope. The choice of the complementary filter in both studies was motivated by specific characteristics of these sensors. The accelerometer tends to be sensitive to high-frequency noise, requiring the implementation of a low-pass filter to mitigate it. On the other hand, the gyroscope suffers from drifting over time due to integration, necessitating a high-pass filter. Considering these factors, the accelerometer readings, expressed in degrees, and the gyroscope rate readings in deg/sec, undergo effective processing through the complementary filter (Ferlini et al., 2019; Odoemelem et al., 2019).

However, tracking head movements using a device that lacks a magnetometer presents a difficult challenge due to the absence of a reference point for sensor recalibration and estimating the 3D orientation of the tracked object (Ferlini et al., 2019). To address the issue arising from the lack of a magnetometer, Ferlini et al. (2019) instruct the study participants to wear a pair of left earbuds during the experiments, then they combine and average data from both earbuds using a 200ms window. The researchers record head movement data in the participants' ideal condition, standing in silence, as a baseline. They then examine how chewing and talking affected the accelerometer and gyroscope signals. To detect head motion, they first switch to the quaternions coordinate system to solve the problem of gimbal lock, then they employed a complementary filter. The findings of this

study reveal that eSense can accurately detect head motions within a few degrees. However, the accuracy decreased when participants engaged in talking or chewing (Ferlini et al., 2019).

While Odoemelem et al., (2019) also utilize a complementary filter for head motion detection, their study diverges in terms of both purpose and approach from that of Ferlini et al. (2019). Odoemelem et al., (2019) present an innovative and economical method for real-time control of a robot arm using head movements. The proposed system by Odoemelem et al. (2019) undergoes several procedures during its operation. The acceleration data captured by the IMU is converted from the earth's reference frame to the IMU body frame. To address unwanted noise and drift in the accelerometer and gyroscope data, filtering techniques like low-pass and high-pass filters are implemented accordingly. Calibration is conducted to accommodate variations in users' individual head orientations, and mapping is utilized to associate the estimated pitch and roll angles with the corresponding pitch and yaw movement of the robot arm (Odoemelem et al., 2019).

Another noteworthy study investigating head movement is conducted by Radhakrishnan et al. (2021). However, in contrast to the previously discussed studies that primarily rely on the complementary filter approach for head motion tracking, Radhakrishnan et al. (2021) implement their own custom algorithm for detecting head motion and incorporate a low-pass filter to mitigate noise interference. Their objective is to identify and rectify inaccurate head positions that may lead to the development of FHP. They illustrate how the gyroscope and accelerometer sensors embedded in earables can effectively recognize instances when the head is excessively tilted forward for extended periods exceeding 30 seconds. Subsequently, the device produces appropriate audio feedback to correct the posture. Through a preliminary study on a limited scale, they demonstrate the potential of their method by achieving exceptional levels of accuracy, with a precision rate of 100% and a recall rate of 89%.

Conversely, Kim et al. (2020) and Purabi et al. (2019) focus on developing machine learning models to achieve the objective of tracking head movement. Purabi et al. (2019) explore monitoring of head movement using earables to examine the relationship between head movement and human attributes as well as emotional conditions. However, they do not delve into the specific technical aspects of machine learning algorithms or the utilization of eSense devices. The researchers adopt a distinct approach by employing the Auto-WEKA package within the Weka software to automatically identify the most suitable classification or regression algorithm based on the provided training dataset (Purabi et al., 2019). In comparison, Kim et al. (2020) provide a more detailed overview of their approach.

Kim et al. (2020) in their study aim to detect head-related behaviors using in-ear IMU data that correlate with online learners' comprehension levels during virtual lectures. Kim et al. (2020) initially seemed to have limited knowledge regarding which head movement behaviors were associated with understanding online lectures. Consequently, they undertook a systematic approach to identify the specific head movement they were seeking to investigate. Kim et al. (2020) discovered that gazing at a monitor and looking down at the desk were the key contexts related to understanding. Subsequently, the researchers developed a machine-learning model using IMU signals from the earbuds to identify behaviors relevant to understanding. They utilized techniques such as feature extraction from the IMU signals in both the time-domain and frequency-domain, as well as employing Principal Component Analysis and a Random Forest Classifier. The performance of the behavior detection model was evaluated through 10-fold cross-

validation. To address imbalanced data distribution, the synthetic minority oversampling technique was applied. The model demonstrated a reasonable detection accuracy, with an average F1 score of 0.79. Furthermore, the study reported specific F1 scores for behaviors of interest, including gazing at a monitor, looking down at the desk, and other identified behaviors (Kim et al., 2020).

Most of the studies reviewed in this section utilize earables to monitor head movement; however, their objectives differ from the objective of this thesis. Specifically, the goal of this thesis is to track head movement for the purpose of driver monitoring. The study conducted by Han et al. (2023) aligns closely with the present study, yet they employ a distinct device; while eSense is explored in this study, they utilize an IMU integrated within a helmet. It is worth noting that there is currently a gap in the research as there are no studies that utilize earables to track head motion for driver monitoring purposes. This gap serves as motivation for this research in this area.

## 2.3 Driver Monitoring

Distraction and lack of focus in vehicle drivers play a crucial role in traffic accidents and road collisions, and there is a growing concern that this problem will increase with the integration of more technologies in vehicles (Regan et al., 2011). Therefore, it is important to develop effective systems for monitoring drivers' behavior.

Smartphones and wearables have gained widespread usage in monitoring drivers, as evidenced by several studies conducted on their effectiveness; see Table 2.

**Table 2**. Systems and devices for driver monitoring in different studies.

| Reference | Proposed System | Device Used |
|---|---|---|
| H. Jiang et al. (2021) | DriverSonar | Smartphones |
| Yu et al. (2017) | $D^3$ | Smartphones |
| Chen et al. (2015) | V-Sense | Smartphones |
| You et al. (2013) | CarSafe | Smartphones |
| Johnson & Trivedi (2011) | Driving Style Recognition | Smartphones |
| Fan et al. (2022) | SafeDriving | Wearables |
| Huang et al. (2019) | MagTrack | Wearables |
| L. Jiang et al. (2018) | SafeDrive | Wearables |

To identify careless driving behaviors, Fan et al. (2022) and Huang et al. (2019) use wearables. However, the two studies adopt different methodologies. While the former in SafeDriving uses data from EMG sensors embedded in smartwatches and develops a deep-learning model to detect abnormal behaviors (Fan et al., 2022), the latter in MagTrack collects data from magnetic tags worn by the car driver and develops models based on analysis and approximation of the collected data to classify the movements of the driver's hands and head (Huang et al., 2019).

Additionally, Jiang et al. (2018) in SafeDrive combine IMU data obtained from a smartphone and wrist-worn device to identify and examine instances of driver distraction. SafeDrive implements a semi-supervised machine learning approach to detect distracting activities within vehicles. To enhance the precision of detection, L. Jiang et al. (2018) incorporate dynamically updated classifiers through the collection of real-time gesture data. Furthermore, they leverage smartphone sensing to generate subtle cues that filter out abnormal movements and non-distracting hand gestures (Jiang et al., 2018).

Similarly, Yu et al. (2017), Chen et al. (2015), and Johnson and Trivedi (2011) employ smartphones to sense diving behaviors, whilst they concentrate more on the vehicle condition. Yu et al. (2017) propose $D^3$ that utilizes accelerometer and gyroscope sensors in smartphones to monitor driving behavior and extract distinctive characteristics, then trains a machine learning algorithm to create a model that can accurately identify and classify abnormal driving behaviors. Chen et al. (2015), on the other hand, designed V-Sense, "a vehicle steering detection middleware", that uses non-vision sensors available on smartphones to identify and distinguish different types of vehicle maneuvers for safety-assistance purposes. Finally, Johnson & Trivedi (2011) present an innovative approach that combines Dynamic Time Warping with sensor fusion on smartphones to identify and capture driving patterns without relying on external processing.

Likewise, H. Jiang et al. (2021) and You et al. (2013) use smartphone sensor data to detect drivers' inattention yet employ different methods. DriverSonar by H. Jiang et al. (2021) is an acoustic-based system that uses the microphone embedded in a smartphone for sensing, while CarSfae by You et al. (2013) is a camera-based system that utilizes front- and rear-facing cameras on smartphones human activity recognition. H. Jiang et al. (2021) argue that high-risk driving behaviors have unique acoustic characteristics, which serve as the foundation for the development of DriverSonar. On the contrary, CareSafe (You et al., 2013) employs computer vision to identify dangerous driving actions, specifically addressing the limitation of smartphones in processing video streams from both front and rear cameras simultaneously. DriverSonar and CarSfae both activate real-time alarms through smartphones when detecting dangerous driving conditions.

Most of the previously discussed studies focused on using smartphones for monitoring driving behavior, while the use of wearables for the same purpose has gained attention only recently. However, the wearables utilized in these studies are predominantly worn on the wrist, such as smartwatches. Notably, no one seems to have explored the use of earables for monitoring driver behaviors. In this study, earables, specifically the eSense device, are employed to monitor and analyze driver behaviors.

# 3. Methodology

The main research methodology of this thesis is case study. More specifically, the approach is a quantitative exploratory case study that aims to study the effect that car movements have on the classification of head orientation. The guidelines provided by Runeson and Höst (2009) were used as the basis for planning and carrying out the research.

The next section details the case study approach. It is followed by the sections addressing the case description, the explored preprocessing approaches, and the included machine learning algorithms, respectively. The description of data preprocessing starts with the applied data cleaning process, which is followed by the investigated data transformation approaches, including the one signal processing method. One section is dedicated for the implementation's technical details. Afterward, the data collection is described. It covers both the data collection that occurred before the thesis work as well as the data collection that was carried out as part of the thesis work. Finally, I conclude this chapter with the selected methods used for the data analysis.

## 3.1 Case Study

As Runeson and Höst (2009) explained Robson's description of the purpose of exploratory case studies, they are about "finding out what is happening, seeking new insights and generating ideas and hypotheses for new research" (Robson, 2002, as cited by Runeson & Höst, 2009, p. 135). This was an applicable method for this research as I sought to find insight into head position tracking when there is noise present in the data. In my case, the noise is brought by the car movements that affect both the accelerometer and gyroscope measurements.

Case study is an empirical method aimed at investigating contemporary phenomena in their context (Runeson & Höst, 2009, p. 134). The method allows for gaining insight into the desired aspects of the targeted case within a realistic context. The benefit of realism comes at the cost of controllability of other variables brought by the context, which reduces the generalizability of the results (Runeson & Höst, 2009). All across the study, data is collected about the relevant aspects of the case so that statistical analyses can be carried out at the end (Wohlin et al., 2003). Although case study can be depicted as a merely observational study (Wohlin et al., 2003), my approach takes a slightly more modern perspective that overlaps with design science research. The overlap comes from the part that in design science research the researcher designs and studies an IT artifact in context (Wieringa, 2014). However, the goal of this thesis is not to provide a new artifact that can solve the problems addressed by the research questions. Rather, the goal is to map and evaluate the effectiveness of machine learning and signal processing solutions in tracking head movements despite the noise created by car movements.

Robson (2002, as cited by Runeson & Höst, 2009) also identified three other purposes for case studies: descriptive, explanatory, and improving. As the name suggests, a descriptive case study aims to characterize the targeted situation or phenomenon. The difference with the explanatory approach is that the explanatory approach is looking for reasons behind and affecting the case. The improving case studies, as expected, seek to provide improvements to the case from a desired perspective (Robson, 2002, as cited by Runeson & Höst, 2009). In addition to these approaches, Klein and Myers (1999, as cited by Runeson & Höst, 2009) detail positivist, critical, and interpretive takes on the

methodology. The positivist case study takes a quantitative empirical approach by forming evidence via measurements and hypothesis testing to generalize results from the case to the target population. The critical case study approach seeks to identify "different forms of social, cultural and political domination that may hinder human ability" (Klein & Myers, 1999, as cited by Runeson & Höst, 2009, p. 135). Finally, the interpretive case study gains insight into the case using the participants' interpretations of their context (Klein & Myers, 1999, as cited by Runeson & Höst, 2009).

According to Runeson and Höst (2009), case studies were initially used mainly for exploration, and the descriptive goals were only targeted whenever the cases were more unique.

## 3.2 Case Description

The case in this case study consists of head movement data collected from eSense earbuds. The movement data is a sequential time series data which includes three axes of accelerometer and three axes of gyroscope. There is so called stationary data that is collected from a person wearing the eSense earbuds, sitting still, and moving their head from left to right and back again. The orientation of the head is labeled into three categories: left, straight, and right using a smartphone application. Similarly, the noisy data is also labeled but the difference is that it is collected from a person driving a car. The movement of the car brings noise to the collected data, which is expected to hinder accurate classification. The noisy data is also more heavily imbalanced as the driver of the vehicle had mostly faced their head forward during the data collection. The data is described in more detail in section 3.6.1 and the results gained directly from this data are explained in section 4.1.

The main use case of the case is to provide an effective head movement tracking solution targeting drivers of vehicles. Starting small by simple classification can be useful for future studies seeking to build more logic on top of the classifiers.

The approach for solving the classification problem is machine learning. This entails selecting machine learning algorithms to be used as well as planning and developing the data cleaning and transformation pipelines. The data cleaning and transformation techniques are discussed in section 3.3, and the selected machine learning algorithms are introduced in section 3.4.

As the amount of collected data is low, it was worth exploring the utilization of k-fold cross validation. As Dalianis (2018) explains it, in k-fold cross validation the dataset is divided into k folds where all but one are used for training and the last one for testing. This is repeated k times so that each fold is used for testing once, and an average is calculated for the results of these folds (Dalianis, 2018).

As part of the data cleaning, I have explored leaving out samples with missing values, dropping timestamp and coordinate attributes to focus on the accelerometer and gyroscope, and undersampling the largest category when using the noisy data. Data transformation efforts include splitting the data into constant length sequences of samples. This allows smoothening of the dataset as it increases the number of samples by a magnitude and decreases the differences between data samples.

## 3.3 Data Preprocessing Approaches

Multiple data cleaning and transformation approaches were experimented with during the development effort. Not all of the approaches described in this section were applied to the data every time, however. For example, it made no sense to balance the noisy data every time before calculating metrics for it using a model trained with the stationary data. The noisy dataset provides a realistic distribution of labels, which the trained models should be able to handle if the model is to have any real-world generalizability. This section will go through the explored data cleaning and transformation methods.

### 3.3.1 Data Cleaning

As the goal of the work was to explore the machine learning models' capability to predict the head orientation based on accelerometer and gyroscope measurements', it made sense to drop the id, longitude, and latitude columns from the data. As the data was imported with sample sequence intact, there was no need to keep the timestamp column either so it was also dropped as part of the data cleaning process. The datasets also contained a lot of missing samples, which could be seen as rows with zeros for all values. These samples were discarded before using the data as they were clear outliers that did not provide any value and there was no clear way to replace them with real values without impacting the realness of the data.

### 3.3.2 Undersampling

To make more generalizable and useful models from the noisy dataset, the data needed to be balanced before training any models with it. The main approach for balancing the dataset was randomized undersampling, which means the size of the dataset was reduced to a total of 6,915 samples with a distribution ratio of 1:1:1 meaning 33% percentage per label. Undersampling was chosen instead of oversampling as the small dataset size could bring an unacceptable amount of bias to the models if samples were directly duplicated meaning the generalizability would suffer a lot. In addition, generating random noise to the duplicated samples would reduce the realism of the data, which in turn might cause the models to learn patterns not visible in real world data. With limited data to validate the models, oversampling was not adopted as an approach to balance the datasets.

Two main approaches were chosen for the data transformation. First, a sequencing algorithm was developed for splitting the data into more consistent sequences. Second, data filtering algorithms were applied to the data both to reduce the dimensions and to highlight the orientational value of the data for the machine learning models to harness.

### 3.3.3 Data Sequencing

The developed sequencing algorithm looped through the data samples, building a list of them until the label changed. Whenever the label changed, the build list was looped through and split to match a desired window length with a desired overlap. Window sizes of 5, 10, 15, and 20, and overlaps of 10 were explored. The usage of overlap turned out to be difficult as the sequencing algorithm depended on the sequentiality of the data, and splitting the data into training and validation sets randomly broke that dependency. The data could not be split after the sequencing as that would mean the overlap brings parts

of the validation set into the training and testing data and the other way around. Therefore, the overlap was not utilized.

In more technical detail, the data was transformed into a three-dimensional Numpy array. However, as the used machine learning algorithms were not able to process this type of data, the dimensions had to be reduced back to two by combining the sequences within the windows of samples. This means the array size of 3267x10x6 (sample sequences times the window size of 10 times the number of columns, which translates to Ax, Ay, Az, Gx, Gy, and Gz) for the stationary data was transformed into 3267x60, where the accelerometer and gyroscope measurements followed each other sample by sample forming one list. The labels were handled separately in another Numpy array to keep the data format complexity low.

### 3.3.4 Complementary Filter

It was expected that machine learning by itself may not be a sufficiently efficient solution for the classification task given the added challenge of potential noise in data. Therefore, the signal processing options were investigated and explored as part of this study. The main signal processing algorithm included in the data transformation process is the complementary filter (Higgins, 1975) which estimates the orientation of the sensor based on sequential gyroscope and accelerometer data.

The filter was adapted into the code in two ways: As a standalone transformation block that works on the data sample by sample and integrated as a configurable option for the sequencing algorithm. However, it was quickly noted that the filter reduced the amount of data drastically and halved the metrics scores, so it was dropped out completely. Either there was an implementation error, or the filter did not work at all with the machine learning algorithms or the data.

## 3.4 Machine Learning Algorithms

Three machine learning algorithms were used for exploring solving the classification task. These are the Random Forest Classifier (RFC) (Breiman, 2001), Logistic Regression (LR) (Cox, 1959), and K-Nearest Neighbour (kNN) (Peterson, 2009). The use of Support Vector Machines (SVM) (Cortes & Vapnik, 1995) was also explored, but the training of the model took unreasonably long, i.e., over one hour, using the available hardware so it was left out of the thesis work for practicality reasons. These algorithms were selected as they are suitable for classification tasks, they are capable of supervised learning, and their implementations were readily available. In addition, at least the SVM, RFC, and kNN algorithms can be considered to be popular shallow classifiers (Lee et al., 2019).

### 3.4.1 Random Forest Classifier

RFCs consist of decision tree classifiers as they use a large collection of the latter to produce predictions for the class, and finally picking the most common one out of the results (Breiman, 2001). However, it is also possible to use random forest for regression (Fawagreh et al., 2014). The decision tree classifiers are tree-structured, and they produce a prediction for a class when given an input (Breiman, 2001). This is handy as the slightly differently built decision trees have different if-else conditions and therefore they can provide different logic for choosing the predicted class. During prediction the tree is

traversed from the top towards the leaves at the bottom, checking the conditions one by one and proceeding with the branch that the condition is true for. Using multiple trees to vote on the best guess for the label offers more varied approaches, which in turn mitigates any singular biases and faults present within the individual trees' classification logic.

The logic behind the random forest classification is straightforward. Each trained decision tree is used to produce a classification, and finally, the most common output is given as the final class. The decision trees work by starting with the top of the tree condition and checking the given sample against it. The next node is selected based on whether the condition is True of False. Reaching a leaf node in the tree means a result has been reached and the class associated with the leaf is the output of the tree.

It must also be considered how these random forests are constructed. This process is depicted in pseudo-code Algorithm 1 (adopted from Fawagreh et al., 2014). In short, the number of trees to be trained for the forest N, the training dataset S, and the features F are given to the function, and it will return the list of trained decision trees, i.e., the random forest. The bootstrap sampling is used to gain variance in the tree generation logic. By considering the features of the chosen sample, the best split feature is determined along with the condition border value, which separates the follow-up branches. By looping through this node building, more and more of the instances are covered by the branches of the tree. When all samples are classified, the tree is complete.

**Algorithm 1.** Random forest formation algorithm (adopted from Fawagreh et al., 2014).

```
Function random_forest(N, S, F):                                    (1)
    RF <- 0
    for i = 1 -> N:
        Ti <- Empty decision tree
        do
            Sample S out of all features F using Bootstrap sampling
            Fs <- Create a vector of the S features
            B(Fs) <- Find best split feature
            Create a new node in Ti using B(Fs)
        while not all instances covered
        Append Ti to RF
  return RF
```

### 3.4.2 Logistic Regression

LR is the oldest of the selected approaches, and there is a lot of literature utilizing it for solving a wide variety of tasks. The algorithm works by estimating the probabilities that a given sample belongs to a class (Géron, 2019, p. 142). The binary classifier form checks whether the likelihood of belonging to the positive class exceeds the threshold of 50% in which case it would be labeled as such. Similarly, reaching a likelihood of below 50% labels the sample as part of the negative class. This point is called the decision boundary (Géron, 2019, p. 142).

In practice, LR works much the same as linear regression in that it computes a weighted sum of the input features except that it provides the results of the regression to a logistic function and returns the results of that. This function is a sigmoid function, and its definition can be seen in Equation 1 (Géron, 2019, p. 142).

**Equation 1.** Logistic function for estimating probability for positive class (Géron, 2019, p. 142).

$$\sigma(t) = \frac{1}{1 + \exp(-t)} \quad (1)$$

For multiclass classification, it is possible to either train multiple binary classifiers or to use softmax regression (Géron, 2019, p. 146). It is also possible to use optimization algorithms such as the one described by Liu & Nocedal (1989) as solvers for LR. Their algorithm improves upon the Broyden-Fletcher-Goldfarb-Shanno algorithm by reducing the overhead caused by memory needs using approximation. The Limited-Memory BFGS is a quasi-Newton method for solving large scale optimization problems (Liu & Nocedal, 1989). This is the default solver used by the Scikit-Learn library, so it is the one that has been used in this work. According to the Scikit-Learn library's documentation, the algorithm is able to handle many kinds of training data at the cost of decreased performance with imbalanced datasets (Pedregosa et al., 2011).

### 3.4.3 K-Nearest Neighbour

The kNN classification algorithm has its roots in 1951 when "Fix and Hodges introduced a non-parametric method for pattern classification" (Peterson, 2009). It has since been improved by numerous people as the formal properties of the algorithm were investigated and defined. In its simplicity, the logic of kNN is that first one sample is picked from each different class to be predicted, and then the training begins by picking more and more new samples. Each new sample gets classified with the label represented by the majority of already classified samples within the K number of nearest samples. The distance to the neighbouring samples is calculated using Euclidian distance in the multidimensional space (Peterson, 2009). The equation for calculating the Euclidian distance in multidimensional space is provided as Equation 2 (Tabak, 2014, p. 150).

**Equation 2.** Calculation for Euclidian distance in multidimensional space (Tabak, 2014).

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \quad (2)$$

## 3.5 Implementation

The machine learning solution was developed using Python 3 in a Jupyter Notebook. The data was imported from CSV files, and the noisy car data and clean sitting still data were separated into different datasets via file name based filtering. All the data files contained missing samples that had a correct sequential ID number but for which all data column values were 0.

The Jupyter Notebook was written to be configurable. One cell was written to contain all of the global constants, which could then be easily altered to adjust which dataset is used, how it is cleaned and transformed, which machine learning algorithms are used, and if the models are going to be trained or if the previously trained models are used. This simplified the process as it was easy to train the models using an undersampled dataset, and then test the models using a different dataset without using undersampling.

Each run of the notebook produced metrics to be collected. The number of samples in the chosen dataset was one of the most common ones. The number of samples depended on whether noisy or stationary dataset was used, if undersampling was applied to it, and how many samples were included in a sequence as part of sequencing if sequencing was used. In addition, each tested model produced many metrics for evaluating the effectiveness of the model on the test data. These data were collected for the analysis of this study.

The data preprocessing was handled using Pandas and Numpy libraries, visualization was done using matplotlib and seaborn, and the machine learning specific algorithms, metrics, and model tuning functions were imported from the Scikit-Learn library. The Scikit-Learn was selected as it is considered easy to use and its implementations of the algorithms are considered efficient (Géron, 2019).

## 3.6 Data Collection

This chapter describes how data collection was carried out in this thesis work. The data collection for the machine learning models took place before the thesis work and it was not carried out by the author of this thesis, so it is described here first. Afterwards, the data collection that was performed as part of the machine learning model development effort is described. This was done by the author of the thesis. The first data collection provided data for the machine learning, and the second data collection provided data for the data analysis of this thesis work.

### 3.6.1 Data Collection for the Machine Learning

The data collection for training and testing the machine learning models occurred before the work was started on this thesis. The data was collected by a different person than the author of this thesis work.

The data was collected using wireless stereo eSense earbuds. Kawsar, Min, Mathur, Montanari, Acer, et al. (2018) describe eSense as “an open and multi-sensory in-ear wearable platform for personal-scale behaviour analytics”. The purpose of eSense is to have the capability of monitoring various head- and mouth-related actions, in addition to observing overall body movement (Kawsar, Min, Mathur, Montanari, Amft, et al., 2018).

There was a total of six participants. Five of them provided data for the ‘stationary’ dataset in which the participants sat still and moved their head between left, straight, right, and back again while wearing the earbuds. The data was sent from the earbuds to a mobile phone via Bluetooth, and they labeled the data in real time by pressing buttons in a mobile application.

The sixth participant was a driver of a car, and they provided the data for the ‘noisy’ dataset. They did not follow the same instructions for turning the head around as the participants for the stationary data but rather their head moved according to the real needs of the traffic. A passenger was present to perform the data labeling during the drive. The driving took place mostly on a Finnish highway, but there were also a few intersections with traffic lights.

The collected data contains 11 columns. The id number is a sequential number looping from 0 to 255 through the samples, unix timestamp, the three axes of the accelerometer (ax, ay, az), the three axes of the gyroscope (gx, gy, gz), latitude, longitude, and finally

the labeled direction for the sample. Due to the duplicative nature of the sequential id numbering, it does not provide any value for the machine learning models. The coordinate data can also be misleading for machine learning as the models should not be dependent on the location. However, it could be used to determine the direction of the movement.

### 3.6.2 Data Collection for the Study

Data was collected by the author of the thesis throughout the study as is customary for case studies (Wohlin et al., 2003). The machine learning solutions are the units of analysis in this case study. Collected data includes result metrics such as Area Under Receiver Operating Characteristic curves, accuracies, recalls, precisions, specificities, F1-scores, and confusion matrices (Dalianis, 2018; Handelman et al., 2019). These metrics are produced for all of the attempted approaches including variations in selected machine learning algorithms, applied data cleaning and transformation efforts.

The machine learning models were trained and tested systematically. A plan was devised to cover different configurations between using noisy and stationary datasets, undersampling, and sequencing with different window sizes. Each trained model was also validated with the 10-fold cross-validation approach. Trained models were also validated with other datasets to see how well they performed with them. For example, a model trained with an undersampled noisy dataset was validated also with the imbalanced noisy dataset as well as both the imbalanced and undersampled stationary datasets. Section 4.2.2 provides more details regarding the different runs of the machine learning code.

Each trained model worked so that with a given input X, a prediction for the label y was produced. In the case of my data, X consists of rows of either six values or a multiplication of it depending on if sequencing is used or not. These six values form the three dimensions for both accelerometer and gyroscope data. The predicted label is the direction. The direction has three options: ‘L’, ‘S’, and ‘R’, representing left, straight, and right respectively.

The data needed to be split into training, testing, and validation sets. 80% of the data was allocated for training the model, and 20% for testing the final model. In addition, as part of the training, 10-fold cross validation was used to produce evaluation metrics scores for each trained model. This means that the training dataset was split into ten equal sized subsets, and each model was trained so that each of these subsets got to be the validation dataset while the others were used for training.

An example of the collected data can be seen in Figure 2. It begins with a generated description of the dataset, which starts with the information gathered from the notebook configuration of the run, and it updates as different notebook cells are run and more preprocessing steps are applied. After the description, there are sample counts and percentages divided label by label so that the distribution and label balance can be seen. If undersampling is applied, both the before and after undersampling statuses are included. The example in Figure 2 contains only the statistics for the RFC model to save space, but every run produced statistics for each of the included models. This section in the data starts with a description of the dataset used to train the model, and finally the test metrics are printed. All of the metrics mentioning ‘10-fold’ utilize 10-fold cross validation. There are also precision, recall, f1-score, and support on a label by label basis so that the effects of dataset imbalances on the results can be estimated more easily.

```
DATASET DESCRIPTION:
Undersampled after sequencing noisy data (car, car1)
with N/As dropped, id, Latitude, Longitude, TimeStamp dropped

TOTAL SAMPLES BEFORE UNDERSAMPLING: 49,518
SAMPLES PER CATEGORY BEFORE UNDERSAMPLING:
S    45,345
L    2,329
R    1,844
SAMPLES PER CATEGORY BEFORE UNDERSAMPLING:
S    91.57276142008966
L    4.703340199523406
R    3.72389838038693
LOWEST  1844
TOTAL SAMPLES: 5,532
SAMPLES PER CATEGORY:
S    1,844
L    1,844
R    1,844
SAMPLES PER CATEGORY:
S    33.33333333333333
L    33.33333333333333
R    33.33333333333333

RFC
clean model (participant1, participant2, participant3, participant4,
participant5) with N/As dropped, id, Latitude, Longitude, TimeStamp dropped

10-fold accuracy: 0.8811
10-fold precision_macro: 0.8814
10-fold precision_micro: 0.8811
10-fold precision_weighted: 0.8814
10-fold recall_macro: 0.881
10-fold recall_micro: 0.8811
10-fold recall_weighted: 0.8811
10-fold f1_score_macro: 0.8804
10-fold f1_score_micro: 0.8811
10-fold f1_score_weighted: 0.8804
10-fold balanced_accuracy: 0.881
10-fold jaccard_macro: 0.7876
10-fold jaccard_micro: 0.7875
10-fold jaccard_weighted: 0.7876

              precision    recall  f1-score   support

           L       0.46      0.24      0.32      1844
           R       0.46      0.72      0.56      1844
           S       0.34      0.30      0.32      1844

    accuracy                           0.42      5532
   macro avg       0.42      0.42      0.40      5532
weighted avg       0.42      0.42      0.40      5532
```

**Figure 2.** An example of collected data.

Finally, data collected for each tested model is accompanied by a visual confusion matrix depicting the actual numbers for how the label predictions took place in relation to the actual labels. When a multiclass model was trained, a single confusion matrix that contained all of the three labels for each axis was produced. In contrast, whenever the one-versus-rest classifier consisting of three binary classifiers was trained, three confusion matrices were produced. Each of these matrices considered one of the labels as the positive label, and all of the rest as the negative label. This is all due to the practical nature of the classifiers, their results, and the ability to map their results into confusion matrices.

## 3.7 Data Analysis

The data analysis of this thesis work is mainly quantitative as the collected data is quantitative. The main part of the analysis consists of the comparison between different model evaluation metrics. The different models can be evaluated directly against each

other as the metrics are the same. It must be noted that a perfect score is often misleading and there is likely a problem with the model. In addition, a lower score does not immediately equal a worse model, as the sample size must also be taken into account. The realism of the dataset also affects the perceived validity of the model. Good results with the stationary dataset do not directly translate into a good model, as the goal of the study is to find characteristics for a model that can generalize to conditions similar to the noisy dataset.

The data analysis also considers the confusion matrices, which say the other half of the story. While evaluation metrics scores provide a view into the effectiveness of a model, the confusion matrix illustrates the mistakes the model makes.

# 4. Results

The results chapter is divided into three parts. The first one describes the insight gained from the data during investigation of the data and its visualization, as well as the development of the approach applied to preprocessing. Later, the results related to the trained machine learning models are explained. The third and final part ties in the findings into answering the research questions.

## 4.1 Data Inspection and Visualization

This subsection reports the initial results from inspecting the dataset both quantitatively in terms of label distribution and by visualizing the different axes of accelerometer and gyroscope against one another.

### 4.1.1 Overview of the Datasets

Table 3 shows the distribution of the samples between the labels both for the stationary and the noisy data. As it turns out, there are almost twice as many samples in the noisy dataset compared to the stationary dataset. The distribution is also more heavily imbalanced in the noisy dataset. The imbalance comes from contextual factors: The driver of the car is facing straight ahead most of the time when they are driving in the traffic. The slight imbalance in the stationary data comes from the practical implementation of how the data was collected. The participants were asked to move their head to face from left to right and back again during the data collection sessions. As the head faces straight forward in between the left and right stages, the activity provides more labels for the transition stage of facing forward in comparison to the two extremities.

**Table 3**. Distribution of labels in the data.

| Dataset | Label | Number of samples | Percentage of samples |
|---|---|---|---|
| Stationary | Straight | **13,793** | **41.3%** |
| | Right | 9,818 | 29.4% |
| | Left | 9,787 | 29.3% |
| | Total | 33,398 | 100.0% |
| Noisy | Straight | **56,682** | **91.6%** |
| | Left | 2,911 | 4.7% |
| | Right | 2,305 | 3.7% |
| | Total | 61,898 | 100.0% |

### 4.1.2 Scatter Matrices

The labeled scatter matrices for accelerometer and gyroscope values (see Figures 3 and 4) highlight the effects of the different axes on each other. Each column pair is drawn into a scatter plot with colours highlighting different labels. This provides insight into label distribution between the different parts of the two-dimensional plane. In an ideal scenario, there could be different clusters in the data so that each color would have its own cluster and there is a clear empty area between the clusters. The classification between the labels would then be as easy as either defining or training an algorithm to classify the samples based on the closest cluster. However, this was not the case with the datasets in question. The labels seem to be scattered all around the value ranges with some slightly different focus in the coloring in some areas visible here and there. As it makes no sense to provide a scatter plot from a column with itself, the data distribution across the value range is drawn for each of the labels spanning from the top left to the bottom right of the scatter matrix chart.

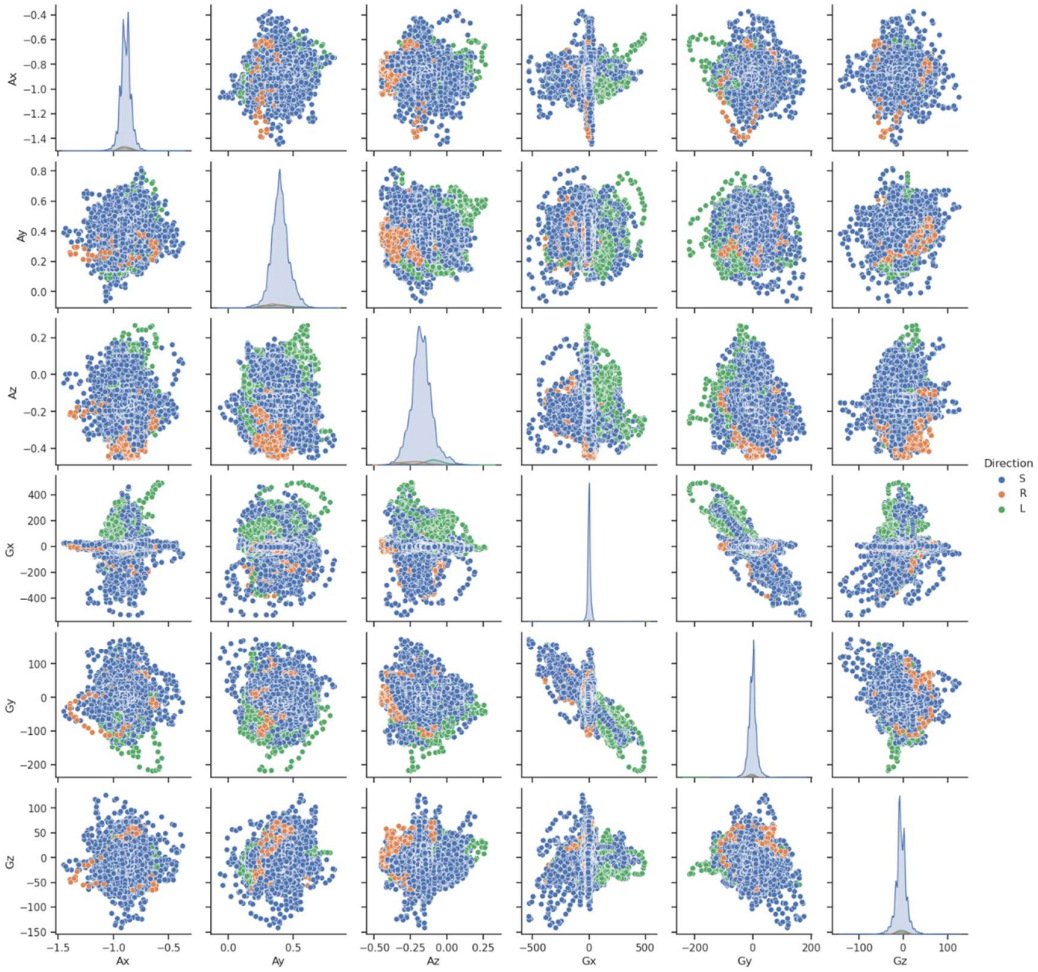


**Figure 3.** Labeled scatter matrix for accelerometer and gyroscope values in the stationary data.

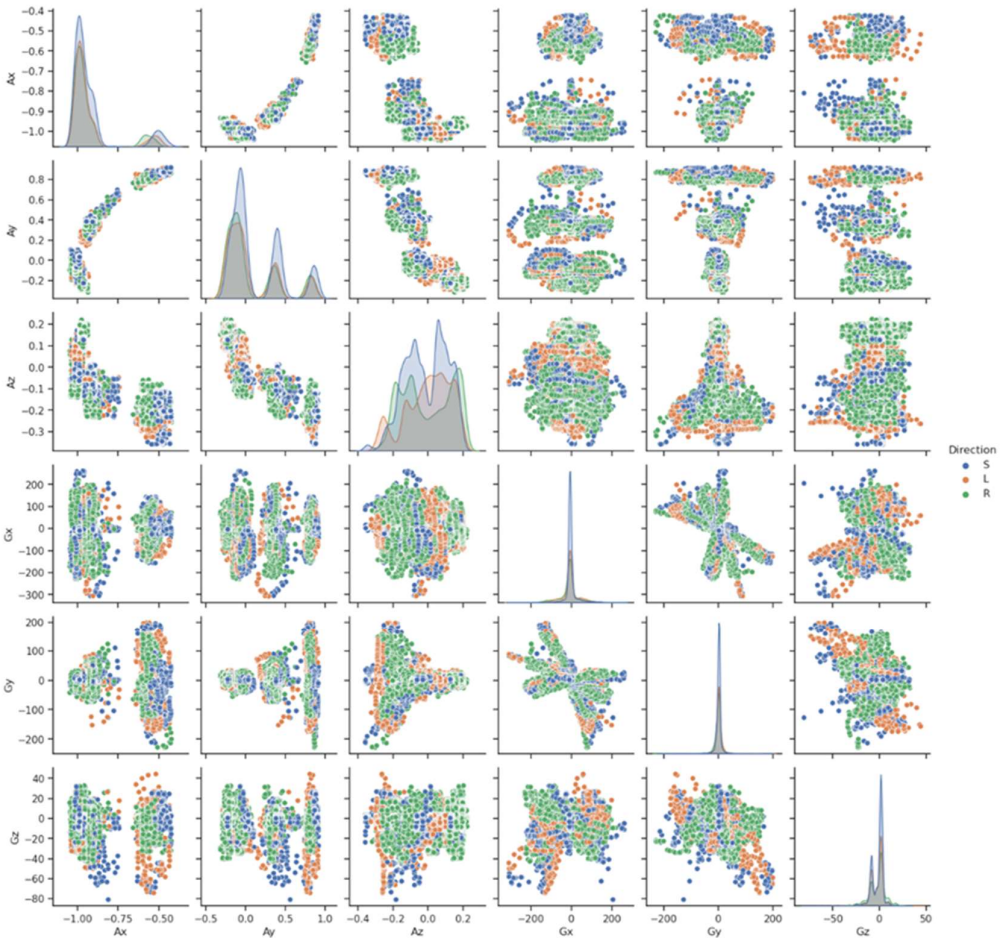


**Figure 4.** Labeled scatter matrix for accelerometer and gyroscope values in the noisy data.

Somewhat interestingly the stationary dataset exhibits a clear dominance of the blue coloring, which portrays the ‘straight’ label. In the noisy dataset, the dominance seems to be on the green ‘right’ label. The previously mentioned ideal case of forming clusters in the scatter matrices is present in some of the plots for the noisy data but the label distribution keeps the same form as all the other ones so there is no clear distinction that one cluster is for a given label and another is for another. For example, the plots between the X and Y accelerometer axes and all of the gyroscope axes seem to portray this pattern of clustering samples into different islands within the plane. The Z axis of the accelerometer does not have such clustering. The accelerometer X axis seems to produce three clusters for each of the gyroscope axes, whereas the accelerometer Y axis produces only two.

Regarding the accelerometer Z axis against the gyroscope X axis, however, it is visible that there is a strand of red spanning across the data cluster indicating a possibility for finding a reasonably simple classification logic recognizing ‘left’ from the other directions. It is also possible to include a third dimension to these scatter plots to better understand the effects of the different axes on each other. These were also explored, but the computational resources required to run interactive 3D scatter plots in a browser caused major performance issues, which hindered analysis too much, and therefore these were left out of the scope of the work.

It is clear from the scatter plots that the X and Y axes of the accelerometer have non-linear correlations in the noisy dataset, but this is not the case for the stationary dataset.

However, in the stationary dataset, there seems to be some linear correlation visible between the gyroscope's X and Y axes.

Overall, the scatter matrices do not provide a clear answer to how the samples could be easily categorized correctly between the labels. There are some minor indications that it might be possible to include a third column for the third dimension so that 2D planes could be drawn to separate different labels from one another with reasonable accuracy.

The results on this part are not certain, and therefore there is no clear answer to be found. However, it must be noted that these matrices only provided the two-dimensional perspective into what kinds of value ranges the columns had in relation to each other. While these value ranges showed that there are both linear and non-linear correlations visible, the label colours pointed out that these correlations do not provide an easy way to predict the directions of the samples using elementary algebra. Fortunately, machine learning models may be able to find patterns from data that are not as clear to the human eye.

## 4.2 Machine Learning Models

This section starts off by explaining issues encountered with splitting the dataset, how these issues were resolved to preface the state of the data given to the machine learning models. Moving on from data preprocessing, general information regarding the trained models is presented, and the rest of the section provides the analysis of the results.

### 4.2.1 Splitting the Dataset

The data preprocessing turned out to be more complicated than was anticipated due to the need to split the dataset. It would be good to split the data into training and testing sets before working with any models so that working on improving the models should not learn any bias that could compromise the generalizability of the final testing. Otherwise, it could be that the models learn to produce highly accurate results both with training and testing data, but they fail to be accurate with other real-world data afterwards. Most of the training data can then be used for training the model, and the rest can be used for model validation. When the models have been fine-tuned to achieve sufficient results, the test set can be used to test the final models.

There was a problem related to sequencing, however. Data could not be split into training and testing data randomly before running the sequencing algorithm as that would break the sequencing logic that depends on the sequential time series nature of the data. Making the split by taking, for example, the desired percentage for the test set from the end of each participant's data could bring bias into the split as the participants may have acted differently towards the end of the data collection. The solution for this problem was to perform sequencing before doing the data split, which complicated data storage, processing, and dataset selection logic in the code as different sequencing window sizes were used.

### 4.2.2 General Information

A total of 44 runs of training and testing on the models were executed for the final evaluation of the effects of the different data cleaning and transformation approaches.

Twelve different models were trained, and they were tested on both datasets with different data preprocessing steps. Two of the test runs only loaded the datasets into memory, deleted N/A values from them, visualized the contents, dropped timestamps and coordinates, and stored backups of them into variables for further runs.

With the small sample size, it is important to consider many factors when assessing a model's performance. The following metrics were calculated for model evaluation: accuracy, precision, recall, f1-score, balanced accuracy, and jaccard scores. More specifically, each of these was calculated using 10-fold validation, and macro, micro, and weighted averaging approaches were all used separately. Micro averaging calculates metrics on the binary confusion matrix formed by adding each class' confusion matrices' values together, macro averaging calculates an unweighted mean for the labels. Finally, the weighted averaging builds on top of the macro approach by considering the label imbalance using label counts as support weights (Pedregosa et al., 2011). In addition, label-wise precision, recall, f1-score, and their support were calculated to provide insight into possible problems with dataset imbalance.

The hyperparameters were not tweaked on any of the models as model fine-tuning was declared out of scope for the work. RFCs were trained with 100 estimators, LR was configured to have a 5,000 as the maximum number of iterations, and the optimal value of K for kNN was searched from [0..100] using mean accuracy of 5-fold cross validation.

### 4.2.3 Analysis

This section contains the analysis of the different machine learning models' performance. The first subsection provides a preface to the analysis by explaining the general results between the models trained with the noisy and the stationary datasets, as well as the structure of the upcoming tables. The second section introduces the most effective models and assesses their performance while comparing the effect of the undersampled dataset to the imbalanced one. The third section brings up the observations related to the effect of different data sequencing algorithm window sizes on the effectiveness of the model. Finally, the section concludes with results related to how the data analysis suffered from the collected misleading metrics and how they were dealt with.

#### 4.2.3.1 Preface to the Analysis

The stationary models and tests on models trained with noisy data did not perform well, lowering the metrics scores close to those of a random classifier, and sometimes even worse than that. This means that only the models trained and tested with noisy datasets performed well enough to be included in the reporting here. They are also well representative of the realistic usage scenarios for which the datasets likely consist of similar data. It should be noted, however, that each sensor is an individual in the sense that there can be minor calibration differences in how a sensor collects data. As such, it can be difficult to generalize a trained model for another sensor if the differences between the sensor model's individuals are too high.

Tables 4 and 5 contain a few preprocessing configurations alongside metrics collected from the model evaluation on testing data. The tables contain a lot of data, but there are some key takeaways to consider here, and the most significant values are bolded to draw attention. It is often easy to reach values close to perfect classification for one label, but it should not come at the cost of the other labels' classification. Attention is given to the

significantly reduced or increased prediction power visible in the metrics. The ‘left’ and ‘right’ labels are given more priority than the ‘straight’ as there is more data for the latter and the former are seemingly more difficult to classify correctly at times.

Let us first consider the format of the tables before taking a look at the insight they hold. The tables display quick summaries of the metrics related to the tested models. Each model trained with undersampled data was also tested with data that had not been undersampled, and the same applies the other way around. This divides the row into two subrows. To get a better view into the possible effects of imbalance on the model, the calculated metrics were divided label by label, which divides both of the sub-rows into three sub-sub-rows. There is only one value per row on the model column as the model was only trained with the first sub-row data, and the trained model was then tested with the both sub-rows’ data one after the other.

#### 4.2.3.2 The Most Effective Models

For each of the included algorithms, Table 4 presents the most effective configurations and the contrasting configuration regarding the utilization of data undersampling for the models. Accuracy is a simple metric to evaluate the models, so let us consider the average of the scores for how the model performed with and without undersampling. It is clear that RFC is the most effective of the three (average accuracy 0.96), followed by kNN (average accuracy 0.83), and LR is the least effective one (average accuracy 0.65). It must be noted that accuracy tells one part of the story, and it is also necessary to consider the label-wise metrics to see how the model manages with the imbalanced dataset. Looking at the lowest label-wise metrics, each model’s lowest metric was the ‘right’ label’s recall (0.59 for RFC, 0.47 for kNN, and 0.11 for LR). Whereas RFC and kNN had better metrics overall for the model trained on an imbalanced dataset, LR benefited from the balanced dataset more.

One noteworthy point is that for RFC and kNN the balance between the trained and tested models’ accuracy remained reasonable (0.01 and 0.15 accuracy loss respectively), but for LR the model trained on the imbalanced dataset performed a lot worse with the undersampled data (0.55 accuracy loss). This is because that model seemed to learn the bias that comes with the imbalance, as can be seen from the model’s incredibly low recall scores (0.01 - 0.02) for the smaller labels.

It seems that both for RFC and kNN it makes sense to work directly with the imbalanced dataset, as the scores remain high even when testing with the undersampled dataset. The precision of the label ‘straight’ had the lowest score for both of the models when tested with the undersampled dataset with values of 0.87 and 0.53 respectively. In contrast, the model trained on the undersampled dataset did not adapt well to the imbalanced dataset as can be seen from the decreased precision score, which went from 0.89 down to 0.31 for label ‘left’ with RFC and from 0.82 to 0.22 for the same label with kNN.

While all of the trained models in the Table 4 have some high scores (each of them achieved at least 0.97 with at least one metric for a label), only the RFC trained on the imbalanced dataset seems to produce acceptable results all around (average between all label-wise precisions and recalls for both datasets at almost 0.90) for reliable classification. The worst of the metrics is the recall for the smaller labels which falls below 0.60, but even this score is at an acceptable level considering a random classifier would produce around 0.33.

**Table 4**. Most effective models for each algorithm.

| Dataset | Under-sampling | Model | Accuracy | Direction | Precision | Recall | F1-score | Support |
|---|---|---|---|---|---|---|---|---|
| Noisy | **No** | **RFC** | .96 | L | .95 | **.61** | .74 | 450 |
| | | | | R | .91 | **.59** | .72 | 348 |
| | | | | S | .97 | 1.0 | .98 | 9,106 |
| | Yes | | .95 | L | **1.0** | .93 | .96 | 1,844 |
| | | | | R | **1.0** | .92 | .96 | 1,844 |
| | | | | S | 0.87 | 1.0 | .93 | 1,844 |
| Noisy | **Yes** | **RFC** | .88 | L | .89 | .92 | .91 | 376 |
| | | | | R | .88 | .94 | .91 | 367 |
| | | | | S | .88 | **.79** | .83 | 364 |
| | No | | .82 | L | **.31** | .97 | **.47** | 2,329 |
| | | | | R | **.32** | .99 | **.48** | 1,844 |
| | | | | S | 1.0 | .81 | .89 | 45,345 |
| Noisy | **No** | **kNN** | .95 | L | .82 | **.50** | .62 | 450 |
| | | | | R | .79 | **.47** | .59 | 348 |
| | | | | S | .96 | .99 | .97 | 9,106 |
| | Yes | | .70 | L | .97 | **.55** | .70 | 1,844 |
| | | | | R | .98 | **.55** | .70 | 1,844 |
| | | | | S | **.53** | 1.0 | .69 | 1,844 |
| Noisy | **Yes** | **kNN** | .80 | L | .82 | .79 | .80 | 376 |
| | | | | R | .78 | .86 | .82 | 367 |
| | | | | S | .81 | .76 | .78 | 364 |
| | No | | .76 | L | **.22** | .82 | **.34** | 2,329 |
| | | | | R | **.24** | .84 | **.38** | 1,844 |
| | | | | S | .99 | .75 | .85 | 45,345 |
| Noisy | **No** | **LR** | .92 | L | .66 | .12 | .20 | 450 |
| | | | | R | .40 | **.01** | **.02** | 348 |
| | | | | S | .92 | 1.0 | .96 | 9,106 |
| | Yes | | **.37** | L | .96 | .10 | .19 | 1,844 |
| | | | | R | .92 | **.01** | **.01** | 1,844 |
| | | | | S | **.34** | 1.0 | .51 | 1,844 |
| Noisy | **Yes** | **LR** | .65 | L | .73 | .68 | .70 | 376 |
| | | | | R | .63 | .70 | .66 | 367 |
| | | | | S | .61 | .58 | .59 | 364 |
| | No | | .57 | L | **.15** | .71 | **.24** | 2,329 |
| | | | | R | **.11** | .71 | **.19** | 1,844 |
| | | | | S | .97 | .56 | .71 | 45,345 |

It is important to also test the models trained on undersampled data without undersampling to see how well they work on realistic label distributions. Overall, models trained with noisy data seemed to perform better on noisy data than on stationary data, and models trained with stationary data seemed to perform better on stationary data than on noisy data.

### 4.2.3.3 Exploring the Sequencing Window Sizes

Splitting the samples in the dataset into sequences, i.e., sequencing, was done with the goal of increasing the effectiveness of the trained models as the amount of data per sample increased. The sequencing algorithm is described in more detail in the implementation section. As the sequencing can be done with different window sizes, there was a need to assess which window size would be the most appropriate. The window sizes were explored by training an RFC model on the imbalanced noisy dataset using sequencing for each included window size. The explored window sizes were 5, 10, 15, and 20. Similarly processed dataset but with undersampling was used to test the model as well. Table 5

shows quantitative data on the effectiveness of different sized sequencing windows for the dataset when using the RFC algorithm.

One important factor noticeable with the different window sizes displayed in Table 5 is that the support values keep going down as the window is increased. This happens because the number of samples decreases significantly as a larger number of dataset samples are required for producing a single sequenced sample.

It also seems that the recall of the 'left' label is linearly decreasing (0.39, 0.31, 0.24, 0.21) as the window size is increased. The 'right' label also decreases from 0.40 to 0.24 with the jump from window size 5 to 10, but there is an increase afterwards to 0.28 and further to 0.29. The accuracy remains more or less the same (max difference of 0.09) for all window sizes. Overall, it seems that the scores are either slowly decreasing or remaining the same as the window size is increased. This is also supported by considering both the lowest (0.39, 0.24, 0.24, 0.21) and the averages (0.71, 0.65, 0.64, 0.60) of label-wise values for precision and recall. These values have been visualized in Figure 5. The trend is clearly mainly negative as the window size is increased. As an additional aspect to consider, the decreased number of samples used for training and verification bring about bias regarding generalizability of the model.

Considering the window size of 5, the model does not seem to be as effective as without the sequencing. The difference in accuracy is 0.02, and the biggest drop is for the label-wise metric for 'left' which drops from 0.61 to 0.39. Gathering multiple samples into one does not seem to be the way to go when improving the effectiveness of the model.

**Table 5**. Sequencing window size exploration.

| Dataset | Under-sampling | Sequencing (Window size) | Model | Accuracy | Direction | Precision | Recall | F1-score | Support |
|---|---|---|---|---|---|---|---|---|---|
| Noisy | No | Yes (5) | RFC | .94 | L | .77 | **.39** | **.52** | 103 |
| | | | | | R | .73 | **.40** | **.52** | 90 |
| | | | | | S | .95 | .99 | .97 | 2,277 |
| | Yes | Yes (5) | | .81 | L | **1.0** | 0.89 | .94 | 455 |
| | | | | | R | **1.0** | .88 | .94 | 455 |
| | | | | | S | 0.81 | 1.0 | .90 | 455 |
| Noisy | No | Yes (10) | RFC | .92 | L | .68 | **.31** | **.42** | 68 |
| | | | | | R | .73 | **.24** | .37 | 45 |
| | | | | | S | .93 | .99 | .96 | 1,120 |
| | Yes | Yes (10) | | .89 | L | 1.0 | .83 | .91 | 224 |
| | | | | | R | 1.0 | .85 | .92 | 224 |
| | | | | | S | 0.76 | 1.0 | .86 | 224 |
| Noisy | No | Yes (15) | RFC | .92 | L | .59 | **.24** | **.34** | 42 |
| | | | | | R | .82 | .28 | .42 | 32 |
| | | | | | S | .93 | .99 | .96 | 747 |
| | Yes | Yes (15) | | .89 | L | 1.0 | .83 | .91 | 148 |
| | | | | | R | 1.0 | .84 | .92 | 148 |
| | | | | | S | 0.76 | 1.0 | .86 | 148 |
| Noisy | No | Yes (20) | RFC | .93 | L | **.45** | **.21** | **.29** | 24 |
| | | | | | R | .70 | .29 | .41 | 24 |
| | | | | | S | .94 | .98 | .96 | 566 |
| | Yes | Yes (20) | | .90 | L | 1.0 | .85 | .92 | 109 |
| | | | | | R | 1.0 | .84 | .92 | 109 |
| | | | | | S | 0.77 | 1.0 | .87 | 109 |

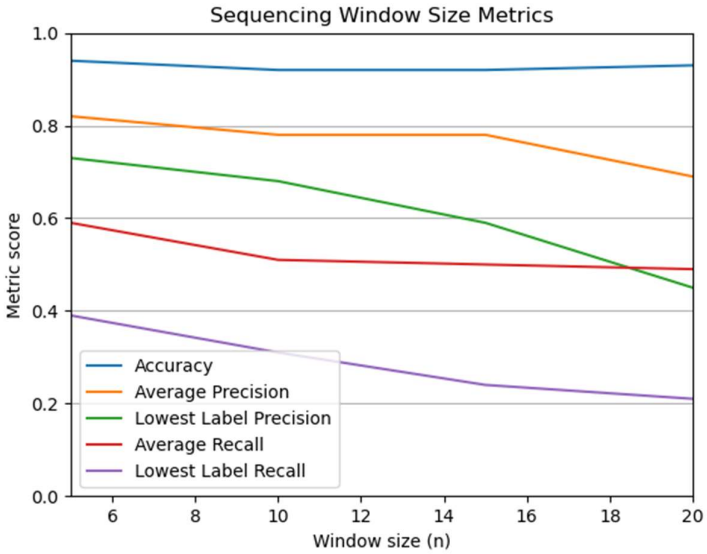


**Figure 5.** Sequencing window size metrics.

The exploration of the window sizes for the data transformation sequencing algorithm provided some insight into their effect on the results. Increasing the window size reduced the number of samples both due to using more of the data for a single sample but also as more of the naturally smaller sequences were dropped with the size increases. The smaller sample sizes lessen the generalizability of the models. Figure 6 shows the confusion matrix of a window size 5 sequenced undersampled noisy data tested on a similarly sequenced imbalanced noisy model. For comparison, Figure 7 shows the exact same situation except for the difference of not using sequencing at all.

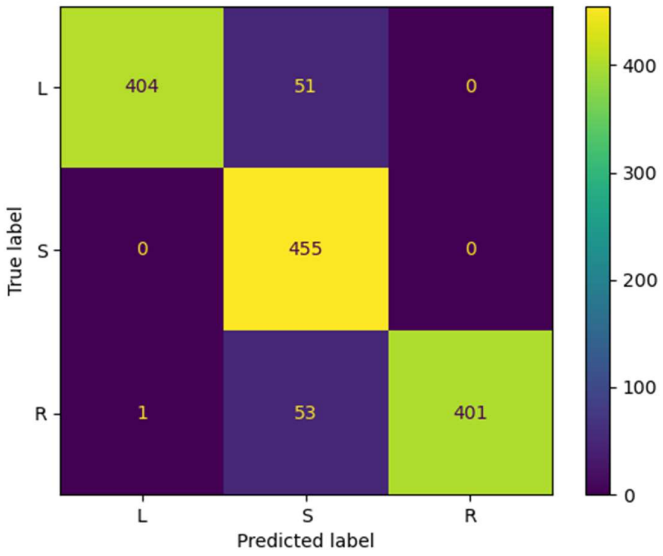


**Figure 6.** RFC with undersampled and sequenced (window size 5) noisy dataset on an imbalanced noisy model.

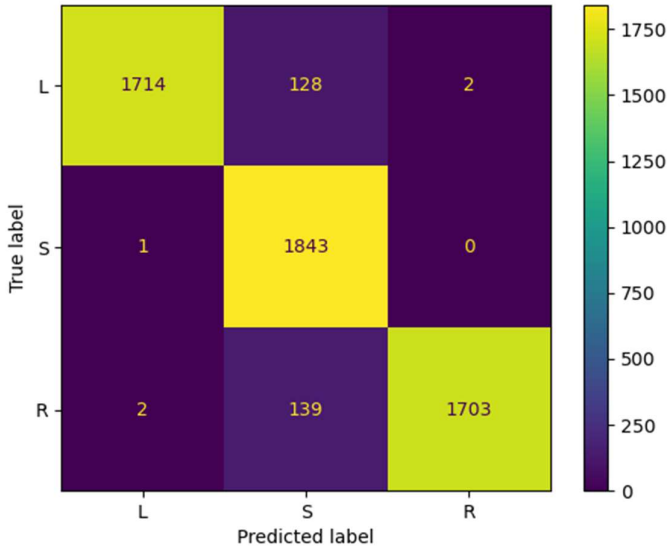


**Figure 7.** RFC with undersampled noisy dataset on an imbalanced noisy model.

### 4.2.3.4 Misleading Metrics and Imbalanced Learning

The data analysis was hampered by some of the collected model performance metrics. It seemed that the metrics present the model's effectiveness as higher than it actually was. It became necessary to consider all collected metrics to gain the overview of how the model actually performs. As per realizing this at an early stage, the data collection was refined to include many different metrics to capture different aspects of the models' performance. This section is dedicated to highlight some of the treacherously one-sided pictures painted by the collected metrics along with other metrics that fill the missing pieces, leading to revealing the real effectiveness of the model.

Often most of the calculated metrics indicated that the trained model was good (with metric values 0.80-0.95), but the label-wise metrics and the confusion matrices told the real result. One example of these misleading metrics is shown as Figure 8. Based on the confusion matrix, the model is clearly not a good one. The 'right' label is predicted correctly well enough (representing 1331 predictions, which is the highest value in the matrix), but there are a lot of misclassifications, e.g., 1040 for 'straight' classified as 'right' and 856 for 'left' classified as 'straight', and the diagonal top-left to bottom-right is not the most represented. Fortunately, the low label-wise metrics raise some suspicion of the model's effectiveness.

Another finding is that models trained with noisy data and tested with stationary data often steered towards predicting straight and left. One example of this is a LR model shown in Figure 9, which classified only one sample as 'right' and even that one was actually 'left'. It was also difficult to assess the validity of models when the testing data was really imbalanced and the confusion matrix displayed the correct classifications as the highest cases, but there are some major misclassifications for the smaller labels. One example of such a case is the RFC model confusion matrix is shown in Figure 10.

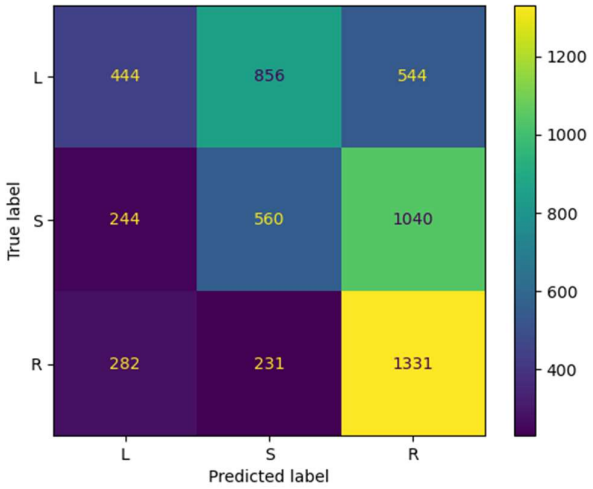


**Figure 8.** An example confusion matrix of a bad model.

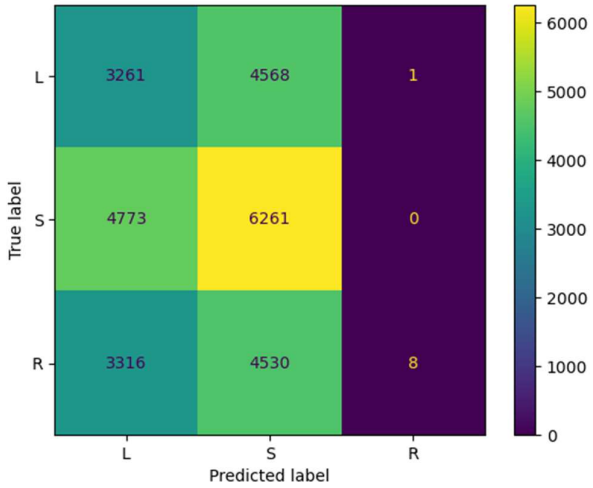


**Figure 9.** LR model neglecting to predict 'Right'.

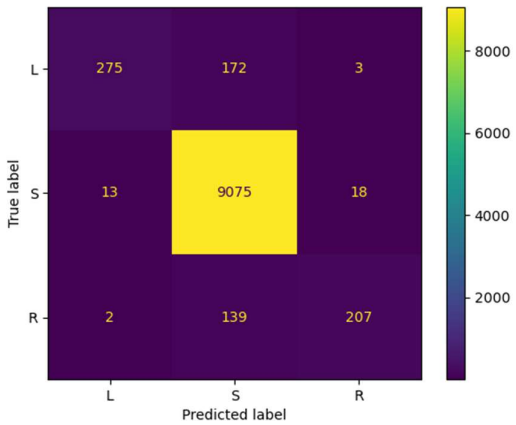


**Figure 10.** RFC tested with an imbalanced dataset.

There were also often generalizability issues related to models trained with the imbalanced noisy dataset as the models learned to predict ‘straight’ and neglecting the two other labels. These models may have fared well against the matching test set, but the results suffered when an undersampled noisy dataset or stationary dataset was used.

## 4.3 Answering the Research Questions

The goal of this study was to explore how head movement data collected from a driver of a car can be used to predict their head orientation in relation to the car. Specifically, the research question is: *How to effectively classify the head orientation of a car driver into left, straight, and right using data from eSense device?* The selected approach of using machine learning to achieve this classification problem raised the question of which algorithms to use and, more importantly, which of them would be the best fit. This question was realized as the first subquestion in the format: *What machine learning algorithm would be suitable for addressing this classification problem?*

As the available data is in the raw format, there was a clear need to clean it before training or testing models with them. Naturally, this need prompted the question of *How to clean the data before utilizing it for machine learning purposes?* This was selected as the second subquestion. The third subquestion was added to consider improving the found solutions by exploring options with the data transformation: *How to transform the data to improve the prediction capabilities of the models?* Together, the answers to these sub-questions provide us with a selection for a machine learning model, steps for cleaning the data before using the model, and suggestions related to how the data could be transformed for increasing the effectiveness of the classification, as desired by the main research question.

**SUB-RQ1.1 What machine learning algorithm would be suitable for addressing this classification problem?**

The first sub question is concerned with which algorithm is the most suitable one for the task. Based on the results, it is clear that out of the algorithms included in the study, the most suitable one is Random Forest Classifier. It produced the most effective models, providing the highest values for the calculated metrics on average. Depending on the data preprocessing steps, the K-Nearest Neighbor algorithm was close to being as suitable as RFC, but most often the kNN algorithm failed to find accurate edges for correct classifications. The LR algorithm performed decently under optimal conditions but failed to generalize at all.

**SUB-RQ1.2 How to clean the data before utilizing it for machine learning purposes?**

The second sub-question is about data cleaning. It should first be noted that the data needs to be read into relevant data structures, which can be provided for the machine learning algorithms. In the case of running Python within a Jupyter Notebook, and using the Scikit-Learn library for machine learning, this means that the data must be transformed into either a Pandas DataFrame or a Numpy Array. Both data formats were explored. The DataFrames could retain more column metadata for the sample by sample model training but they did not allow for two-dimensional arrays. To sequence the data into these two-dimensional arrays, Numpy Arrays were used. Three-dimensional arrays would have been optimal for representing the sequences of samples within the data, but there seems to be no way to achieve this using the tools used in this work.

As for the exact answer to the question, there was little need to clean the data before using it for training the models and predicting with them. It was only necessary to delete the rows with missing data. To make the models focus on the meaningful pieces of data, the columns ID, timestamp, latitude, and longitude were dropped. This allowed the models to only see the accelerometer and gyroscope values, which reduced the dimensions of the data to an acceptable level. It was possible to train the models with a reasonable amount of resources, and the model accuracies were adequate. Improvements could be made to the models' effectiveness by undersampling the data before training the models with them, but this was not necessary for producing good models.

**SUB-RQ1.3 How to transform the data to improve the prediction capabilities of the models?**

The last sub-question considers improving the models' prediction power via data transformation. A complementary filter was tested but it had such large negative effects on the metrics scores it was not considered a viable option past that.

After experimenting with data sequencing and different window sizes, it seems that the grouping of samples into sequences does not affect the models' prediction accuracies. Due to the small sample size, larger sequence window sizes led to even smaller training and testing datasets, which reduces the credibility of the models' generalizability. The metrics scores seemed to go down as the window size was increased, but the confusion matrices maintained mostly the same proportions and distributions. The window size of five data samples per training dataset sample provided the most optimal models with a reasonable amount of generalizability. However, the non-sequenced models seemed to provide similar if not better results overall. Sequencing might have a more positive impact on the models if the dataset was larger as then the reduced number of samples would become more negligible.

**RQ1 How to effectively classify the head orientation of a car driver into left, straight, and right using data from eSense device?**

With the sub-questions answered, I can conclude that the head orientation of the driver of a moving vehicle can be classified using machine learning algorithms trained with the data from eSense wearables worn by the driver. More specifically, the collected data should contain accelerometer and gyroscope values, and missed values should be deleted. Splitting the data into sequences might improve the models' prediction power, but it can be difficult to sequence real-world data into sequences so that all included samples have the same expected label. In other words, sequencing real-world data in real-time will likely end up producing sequences that contain samples from moments in time where the driver's head is facing different directions, yet the sequences should be labeled with only one direction.

# 5. Discussion

The discussion section is divided into four subsections. The first subsection gathers up challenges faced in conducting this study. The second subsection highlights the gap in the literature paving the way for the third subsection which answers how this thesis contributes to filling the gap. The discussion section concludes with the limitations of this work.

## 5.1 Challenges

The data collection process for building the dataset of this study involved participants wearing eSense earbuds to track their head movement while seated and during driving. However, one of the primary challenges encountered in this study was the lack of direct monitoring during data collection. This introduced a significant limitation as the researcher of this study had limited control over the collection process. Inconsistencies and variations in data collection procedures could have occurred, potentially impacting the overall quality and reliability of the data. Without direct oversight, it was challenging to ensure that participants followed consistent protocols, such as maintaining stable head positions during data acquisition or accurately labeling their head movements. Consequently, uncertainties may arise regarding the validity and accuracy of the collected data, which may affect what the models learn from the training data.

Another significant challenge associated with data collection conducted without direct monitoring was the potential introduction of biases and subjectivity. In this study, labeling of head movement was performed by individuals who were not under the direct supervision of the researcher. This introduced the possibility of variations in labeling criteria and interpretations among different labelers. Each labeler may have applied their own judgment and criteria, leading to inconsistencies in the categorization of head movement. This subjectivity may have influenced the overall analysis and interpretation of the data.

As part of the development effort, there were several concerns related to the development of the sequencing algorithm and splitting the dataset into training, testing, and validation datasets. As previously described, there was the problem of sequencing the data into windows with overlaps. The sequential nature of the data required that the data could not be split randomly into the different datasets, or the sequences would break, and splitting afterward could mean that overlap brings the same exact samples into multiple datasets which introduces a bias. With the increasing overall complexity of the sequencing algorithm, some bugs were also introduced. Fortunately, these bugs caused major shifts in the label distributions and the accuracies of the trained models, which raised suspicions that led to fixing the algorithm implementation.

## 5.2 Earables in Traffic Safety: An Emerging Frontier

The concept of earables is a fairly recent one, and it seems to have entered academic literature during the past few years (Powar & Beresford, 2019). Therefore, there is still plenty to explore with different application domains for them. Many studies have already used machine learning to classify situations and activities and also signal processing techniques (Ferlini et al., 2019; Odoemelem et al., 2019) to track head orientations.

Application domains such as ergonomics, healthcare, and navigation have already seen studies related to them. However, there were not many papers discussing traffic safety. It seems that the work on adopting the earables for improving traffic safety has not seen a lot of interest as of yet. In a way, this study is really contributing to the existing research in this domain.

## 5.3 Contribution

The results of this thesis work support the notion that it is not necessary to apply signal processing to reduce the noise present in the dataset collected in moving vehicles. It seems that the machine learning algorithms are able to find the patterns even from the noisy data and to provide acceptable metrics scores for the classifiers. Out of the included machine learning algorithms, RFC gains yet another study supporting its capability to produce effective models for this type of classification task. This result may be part of steering the future research into leaving out algorithms such as the LR from the approaches to explore. Overall, the results are in agreement with previous research, and therefore supportive of the current scientific knowledge and understanding of the topic.

## 5.4 Limitations

As the study was conducted on a particular case, there is some loss level of control which impacts the generalizability of the results (Wohlin et al., 2003; Runeson & Höst, 2009).

The size of the dataset used for training and testing the models was small, and it got reduced even more as undersampling was applied to balance the data. The achieved evaluation scores of the models could have been improved by further fine-tuning the models via hyperparameter optimizations. This limits the potential of the outcome but provides some generalizability of the results as the trained and tested models are more comparable as the hyperparameters remain the same across the test runs.

The lack of knowledge regarding the gender distribution of the participants in the data collection process raises an important consideration. Considering gender as a potential factor influencing head movement patterns is crucial, as it may introduce variability in the data. The lack of information regarding gender distribution limits a comprehensive understanding of the influence of gender on the findings. Future studies should aim to collect data with proper gender representation to account for these potential effects.

# 6. Conclusion

The goal of this thesis work was to contribute to improving traffic safety by facilitating future research on head movement tracking of drivers of vehicles. Informed by the previous research, an approach was chosen for applying a machine learning pipeline to clean and transform the dataset into a format that can be used to train effective models. The primary research question was to find an effective way to classify the head orientation of a car driver into left, straight, and right.

Dropping of missing values and unnecessary columns, undersampling, and sample sequencing were explored as part of the data cleaning and transformation effort. Random Forest Classifier, K-Nearest Neighbor, and Logistic Regression classifiers were trained and evaluated for solving the classification task. One of the main findings is that the RFC performs the best out of the tested algorithms for this classification task given the datasets and tested data preprocessing options. k-NN achieved the second-best results closely following RFC's metrics, and Logistic Regression provided mostly unusable classifiers except for ideal scenarios in which they operated reasonably.

Based on the results it is proposed to train RFC directly on the labeled imbalanced data from the vehicles. Undersampling the dataset to provide a balanced label distribution seemed to decrease the effectiveness of the model. Sequencing was also considered to be both difficult to implement in a real environment and to decrease the effectiveness of the predictions. The sequencing showed some promise with comparable results to sample-by-sample classification, so it might be that with a larger dataset, the results on the effectiveness of sequencing could have been different.

The effectiveness of the RFC trained with a heavily imbalanced noisy dataset was an interesting finding that highlights how the algorithm is able to learn meaningful patterns for correctly classifying even the smallest of labels. It would not be difficult to achieve high metrics with any classifier on such data but achieving reasonable metrics when considering the smaller labels' metrics is not always possible.

This study joins the previous research with the common characteristic of lacking a comprehensive dataset. As such, the results should be taken with the consideration of lowered generalizability, and future research should seek to gather up a more extensive dataset before attempting to replicate or improve upon the results of the previous studies. The effectiveness of sequencing could also be re-evaluated with a more favorably sized dataset.

In terms of broadening the scope, future research could consider carrying out model refinement via hyperparameter tuning and moving from left, straight, and right classification into a two-dimensional coordinate space for displaying the head orientation. It could also be worth exploring if more accurate and robust results would be reached if the problem of head orientation tracking in a moving vehicle could be solved using physics and math instead of machine learning. The future research could also embark on building on top of the head orientation detection models to start the work on driver condition recognition.

# References


Ahuja, A., Ferlini, A., & Mascolo, C. (2021). PilotEar: Enabling In-ear Inertial Navigation. *UbiComp/ISWC 2021 - Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*. https://doi.org/10.1145/3460418.3479326

Aurélien Géron. (2019). Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems. In *O'Reilly Media*.

Bi, S., Wang, T., Tobias, N., Nordrum, J., Wang, S., Halvorsen, G., Sen, S., Peterson, R., Odame, K., Caine, K., Halter, R., Sorber, J., & Kotz, D. (2018). Auracle: Detecting Eating Episodes with an Ear-mounted Sensor. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *2*(3). https://doi.org/10.1145/3264902

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1). https://doi.org/10.1023/A:1010933404324

Cao, G., Yuan, K., Xiong, J., Yang, P., Yan, Y., Zhou, H., & Li, X. Y. (2020). EarphoneTrack: Involving earphones into the ecosystem of acoustic motion tracking. *SenSys 2020 - Proceedings of the 2020 18th ACM Conference on Embedded Networked Sensor Systems*. https://doi.org/10.1145/3384419.3430730

Chen, D., Cho, K. T., Han, S., Jin, Z., & Shin, K. G. (2015). Invisible sensing of vehicle steering with smartphones. *MobiSys 2015 - Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. https://doi.org/10.1145/2742647.2742659

Choudhury, R. R. (2021). Earable Computing: A New Area to Think about. *HotMobile 2021 - Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications*. https://doi.org/10.1145/3446382.3450216

Clarke, C., Ehrich, P., & Gellersen, H. (2020). Motion Coupling of Earable Devices in Camera View. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3428361.3428470

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3). https://doi.org/10.1023/A:1022627411411

Cox, D. R. (1959). The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, *21*(1). https://doi.org/10.1111/j.2517-6161.1959.tb00334.x

Dalianis, H. (2018). Clinical text mining: Secondary use of electronic patient records. In *Clinical Text Mining: Secondary Use of Electronic Patient Records*. Springer International Publishing. https://doi.org/10.1007/978-3-319-78503-5

Fan, Y., Gu, F., Wang, J., Wang, J., Lu, K., & Niu, J. (2022). SafeDriving: An Effective Abnormal Driving Behavior Detection System Based on EMG Signals. *IEEE Internet of Things Journal*, *9*(14). https://doi.org/10.1109/JIOT.2021.3135512

Fawagreh, K., Gaber, M. M., & Elyan, E. (2014). Random forests: From early developments to recent advancements. *Systems Science and Control Engineering*, *2*(1). https://doi.org/10.1080/21642583.2014.956265

Ferlini, A., Montanari, A., Mascolo, C., & Harle, R. (2019). Head Motion Tracking Through in-Ear Wearables. *Proceedings of the 1st International Workshop on Earable Computing, EarComp 2019*, 8–13. https://doi.org/10.1145/3345615.3361131

Gao, Y., Wang, W., Phoha, V. V., Sun, W., & Jin, Z. (2019). EarEcho: Using Ear Canal Echo for Wearable Authentication YANG. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *3*(3).

Han, Z., Xu, L., Dong, X., Nishiyama, Y., & Sezaki, K. (2023). HeadMon: Head Dynamics Enabled Riding Maneuver Prediction. *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*.

Handelman, G. S., Kok, H. K., Chandra, R. V., Razavi, A. H., Huang, S., Brooks, M., Lee, M. J., & Asadi, H. (2019). Peering into the black box of artificial intelligence: Evaluation metrics of machine learning methods. *American Journal of Roentgenology*, *212*(1). https://doi.org/10.2214/AJR.18.20224

Higgins, W. T. (1975). A Comparison of Complementary and Kalman Filtering. *IEEE Transactions on Aerospace and Electronic Systems*, *AES-11*(3). https://doi.org/10.1109/TAES.1975.308081

Hossain, T., Islam, M. S., Rahman Ahad, M. A., & Rahman Ahad, M. A. (2019). Human activity recognition using earable device. *UbiComp/ISWC 2019- - Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, 81–84. https://doi.org/10.1145/3341162.3343822

Huang, H., Chen, H., & Lin, S. (2019). MagTrack: Enabling safe driving monitoring with wearable magnetics. *MobiSys 2019 - Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. https://doi.org/10.1145/3307334.3326107

Jiang, H., Hu, J., Liu, D., Xiong, J., & Cai, M. (2021). DriverSonar: Fine-Grained Dangerous Driving Detection Using Active Sonar. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *5*(3). https://doi.org/10.1145/3478084

Jiang, L., Lin, X., Liu, X., Bi, C., & Xing, G. (2018). SafeDrive: Detecting Distracted Driving Behaviors Using Wrist-Worn Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *1*(4).

Johnson, D. A., & Trivedi, M. M. (2011). Driving Style Recognition Using a Smartphone as a Sensor Platform. *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. https://doi.org/10.1109/ITSC.2011.6083078

Katayama, S., Mathur, A., Okoshi, T., Nakazawa, J., & Kawsar, F. (2019). Demo: Situation-aware conversational agent with kinetic earables. *MobiSys 2019 - Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, 657–658. https://doi.org/10.1145/3307334.3328569

Katayama, S., Mathur, A., Van Den Broeck, M., Okoshi, T., Nakazawa, J., & Kawsar, F. (2019). Situation-Aware Emotion Regulation of Conversational Agents with Kinetic Earables. *2019 8th International Conference on Affective Computing and Intelligent Interaction, ACII 2019*. https://doi.org/10.1109/ACII.2019.8925449

Kawsar, F., Min, C., Mathur, A., Montanari, A., Acer, U. G., & Van den Broeck, M. (2018). Demo abstract: eSense - Open earable platform for human sensing. *SenSys 2018 - Proceedings of the 16th Conference on Embedded Networked Sensor Systems*, 371–372. https://doi.org/10.1145/3274783.3275188

Kawsar, F., Min, C., Mathur, A., Montanari, A., Amft, O., & Van Laerhoven, K. (2018). Earables for personal-scale behavior analytics. *IEEE Pervasive Computing*, *17*(3). https://doi.org/10.1109/MPRV.2018.03367740

Kim, D., Min, C., & Kang, S. (2020). Towards recognizing perceived level of understanding for online lectures using earables: Poster abstract. *SenSys 2020 - Proceedings of the 2020 18th ACM Conference on Embedded Networked Sensor Systems*. https://doi.org/10.1145/3384419.3430428

Kim, D., Min, C., & Kang, S. (2021). Towards Automatic Recognition of Perceived Level of Understanding on Online Lectures using Earables. *UbiComp/ISWC 2021 - Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*. https://doi.org/10.1145/3460418.3479323

Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly: Management Information Systems*, *23*(1). https://doi.org/10.2307/249410

Laporte, M., Baglat, P., Gashi, S., Gjoreski, M., Santini, S., & Langheinrich, M. (2021). Detecting Verbal and Non-Verbal Gestures Using Earables. *UbiComp/ISWC 2021 - Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*. https://doi.org/10.1145/3460418.3479322

Lee, S., Min, C., Montanari, A., Mathur, A., Chang, Y., Song, J., & Kawsar, F. (2019). Automatic smile and frown recognition with kinetic earables. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3311823.3311869

Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, *45*(1–3). https://doi.org/10.1007/BF01589116

Lotfi, R., Tzanetakis, G., Eskicioglu, R., & Irani, P. (2020). A comparison between audio and IMU data to detect chewing events based on an earable device. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3396339.3396362

Matsumura, K., & Okada, K. (2019). ESense veers: A case study of acoustical manipulation in walking without sight both on subtle and overt conditions. *Proceedings of the 1st International Workshop on Earable Computing, EarComp 2019*. https://doi.org/10.1145/3345615.3361135

Min, C., Mathur, A., & Kawsar, F. (2018a). Exploring audio and kinetic sensing on earable devices. *WearSys 2018 - Proceedings of the 4th ACM Workshop on Wearable Systems and Applications*. https://doi.org/10.1145/3211960.3211970

Min, C., Mathur, A., & Kawsar, F. (2018b). Poster: Audio-kinetic model for automatic dietary monitoring with earable devices. *MobiSys 2018 - Proceedings of the 16th ACM International Conference on Mobile Systems, Applications, and Services*. https://doi.org/10.1145/3210240.3210810

Min, C., Montanari, A., Mathur, A., Lee, S., & Kawsar, F. (2018). Cross-modal approach for conversational well-being monitoring with multi-sensory earables. *UbiComp/ISWC 2018 - Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2018 ACM International Symposium on Wearable Computers*. https://doi.org/10.1145/3267305.3267695

Odoemelem, H., Holzemann, A., & Van Laerhoven, K. (2019). Using the eSense wearable earbud as a light-weight robot arm controller. *Proceedings of the 1st International Workshop on Earable Computing, EarComp 2019*. https://doi.org/10.1145/3345615.3361138

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*.

Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, *4*(2), 1883. https://doi.org/10.4249/SCHOLARPEDIA.1883

Powar, J., & Beresford, A. R. (2019). A Data Sharing Platform for Earables Research. *Proceedings of the 1st International Workshop on Earable Computing, EarComp 2019*. https://doi.org/10.1145/3345615.3361139

Prakash, J., Yang, Z., Wei, Y. L., & Choudhury, R. R. (2019). STEAR: Robust Step Counting from Earables. *Proceedings of the 1st International Workshop on Earable Computing, EarComp 2019*. https://doi.org/10.1145/3345615.3361133

Purabi, S. A., Rashed, R., Islam, M. M., Uddin, M. N., Naznin, M., & Al Islam, A. B. M. A. (2019). As you are, so shall you move your head: A system-level analysis between head movements and corresponding traits and emotions. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3362966.3362985

Radhakrishnan, M., & Misra, A. (2019). Can Earables Support Effective User Engagement during Weight-Based Gym Exercises? *Proceedings of the 1st International Workshop on Earable Computing, EarComp 2019*. https://doi.org/10.1145/3345615.3361132

Radhakrishnan, M., Misra, K., & Ravichandran, V. (2021). Applying “Earable” Inertial Sensing for Real-time Head Posture Detection. *2021 IEEE International Conference*

*on Pervasive Computing and Communications Workshops and Other Affiliated Events, PerCom Workshops 2021*. https://doi.org/10.1109/PerComWorkshops51409.2021.9430988

Regan, M. A., Hallett, C., & Gordon, C. P. (2011). Driver distraction and driver inattention: Definition, relationship and taxonomy. *Accident Analysis and Prevention*, *43*(5). https://doi.org/10.1016/j.aap.2011.04.008

Robson, C. (2002). Real World Research: A Resource for Social Scientists and Practitioner-Researchers (2nd ed.). Oxford: Blackwell Publishers Ltd. In *Emergency Nurse* (Vol. 5, Issue 7).

Röddiger, T., Clarke, C., Breitling, P., Schneegans, T., Zhao, H., Gellersen, H., & Beigl, M. (2022). Sensing with Earables: A Systematic Literature Review and Taxonomy of Phenomena. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *6*(3). https://doi.org/10.1145/3550314

Roddiger, T., Wolffram, D., Laubenstein, D., Budde, M., & Beigl, M. (2019). Towards Respiration Rate Monitoring Using an In-Ear Headphone Inertial Measurement Unit. *Proceedings of the 1st International Workshop on Earable Computing, EarComp 2019*. https://doi.org/10.1145/3345615.3361130

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, *14*(2). https://doi.org/10.1007/s10664-008-9102-8

Srivastava, T., Khanna, P., Pan, S., Nguyen, P., & Jain, S. (2022). Leveraging earables for unvoiced command recognition. *MobiSys 2022 - Proceedings of the 2022 20th Annual International Conference on Mobile Systems, Applications and Services*. https://doi.org/10.1145/3498361.3538665

Tabak, J. (2014). *Geometry: The Language of Space and Form*. Infobase Publishing. https://books.google.fi/books?hl=en&lr=&id=r0HuPiexnYwC&oi=fnd&pg=PP1&dq=Geometry:+The+Language+of+Space+and+Form&ots=JLnc7gSp4b&sig=zP4Fbl1GBJ0WEAizayj3U7f-Gi8&redir_esc=y#v=onepage&q=Geometry%3A%20The%20Language%20of%20Space%20and%20Form&f=false

Wieringa, R. J. (2014). Design science methodology: For information systems and software engineering. In *Design Science Methodology: For Information Systems and Software Engineering*. https://doi.org/10.1007/978-3-662-43839-8

Wohlin, C., Höst, M., & Henningsson, K. (2003). Empirical research methods in software engineering. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2765*. https://doi.org/10.1007/978-3-540-45143-3_2

Xu, X., Shi, H., Yi, X., Liu, W. J., Yan, Y., Shi, Y., Mariakakis, A., Mankoff, J., & Dey, A. K. (2020). EarBuddy: Enabling On-Face Interaction via Wireless Earbuds. *Conference on Human Factors in Computing Systems - Proceedings*. https://doi.org/10.1145/3313831.3376836

You, C. W., Lane, N. D., Chen, F., Wang, R., Chen, Z., Bao, T. J., Montes-de-Oca, M., Cheng, Y., Lint, M., Torresani, L., & Campbell, A. T. (2013). CarSafe App: Alerting

drowsy and distracted drivers using dual cameras on smartphones. *MobiSys 2013 - Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services*. https://doi.org/10.1145/2462456.2465428

Yu, J., Chen, Z., Zhu, Y., Jennifer Chen, Y., Kong, L., & Li, M. (2017). D 3: Abnormal Driving Behaviors Detection and Identification using Smartphone Sensors. *IEEE Transactions on Mobile Computing*, *16*(8). https://doi.org/10.1109/TMC.2016.2618873

Zhu, P., Zou, Y., Li, W., & Wu, K. (2023). CHAR: Composite Head-body Activities Recognition with A Single Earable Device. *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*.