UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Assessing road quality and driver behavior using movement sensors in wrist worn devices

Zwet, Dennis van der

Master Thesis
September 2020

Supervisors:
prof. dr. ir. P.J.M. Havinga
ir. J. Scholten
dr.ir. B.J. van der Zwaag
ir. E. Molenkamp

Pervasive Systems Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Summary

In current society, people are involved in traffic every day. In order to travel to go to work, do the groceries, meet friends, study, visit a forest, in all these situations it is necessary to be involved in traffic. Because of this, it is very important that traffic is as safe as it can possibly be. Unsafe situations can be caused by many factors. An important factor is road quality: poorly maintained roads can cause damage to vehicles and even be a cause for accidents. Moreover, the way the road has been laid out can be very important for safety. Hard breaking is a potential cause for accidents. This means that roads that require hard breaking can be a cause for accidents to happen. The fact that it is difficult and labor intensive to check every road for damage and unsafe situations has become problematic. An automated way to detect these situations by people driving on the road would be a possible solution to increase awareness of which roads maintenance.

Another important cause of accidents is driver behaviour. Distracted driving is one of the most important causes of accidents. However, it is easy to be distracted while driving. An automated system to notify a driver that they are being distracted could be beneficial for traffic safety.

To improve these two causes of accidents, a solution must be created to automatically detect road safety, as well as driver behaviour. Therefore, this paper tries to answers two questions:

- Can we detect unsafe situations on the road using smartwatches?
- Can we detect the behavior of a driver using smartwatches?

This paper proposes using sensors already available in wrist worn devices to measure both road safety and driver behaviour. Based on a literature review of existing solutions, this paper investigates three distinct machine learning techniques: Support Vector Machines, Long Short-Term Memory model and the InceptionTime model.

Existing options are surveyed, after which the paper describes its own solution to the problems. The conclusion of the paper is that current methodology needs more improvements to create a viable solution.

Contents

Chapter 1

Introduction

Everyone has to participate in traffic almost every day. People participate for example by walking, cycling or driving a car or other motorized vehicle. Because of this high exposure to traffic, safety on the road is an important factor for the overall safety of people. However, too many accidents still happen on the road, with various causes. Most accidents involve a certain degree of human error. Some risk factors increasing or decreasing the chance of human error can be influenced by the road infrastructure.

One factor is the quality of road pavement, which can cause accidents by damaging the car, or requiring drivers to evade certain patches of roads, increasing the risk of hitting something else. Another factor is how the road is laid out. One can think of factors such as the width of a road, or how easy it is to see upcoming traffic at an intersection. Also, the speed difference on converging roads can directly influence traffic safety. In the area of Civil Engineering, serveral guidelines have been laid out for safe roads [1] [2] [3] [4] [5] [6] [7] .

Another important aspect in traffic safety is driver behavior. The amount of aggression that someone drives with, or how attentive someone is, directly influences the chance of colliding. However, also distractions like listening or using the radio, having a conversation with a passenger sitting in a car, or even opening a window can increase the chance of an accident.

Most of these risk factors can be monitored. There are expensive machines that can detect every irregularity in the road. Research can be done into certain intersections, on how they can be made safer. Hazardous driving behavior can be spotted by traffic cops. However all these options require large investments of time and funds in either expensive machinery or employees.

Nowadays, the availability of devices equipped with potent sensors is relatively high. This implicitly means that many participants in traffic carry one or more sensing device. Multiple researchers have identified the possibility to use those sensors to measure certain aspects of road safety.

In a previous paper [8] we identified multiple areas where research has been lacking. We identified if it would be possible to detect unsafe situations in roads, for example by measuring whether certain areas have very tight turns or hard breaking occurs at certain spots. Another area is monitoring driver behaviour without requiring extensive setups for example involving cameras.

We think both areas can be solved using wrist worn consumer electronics. We expect wrist worn devices with movement and position sensors can be harnessed to detect what a driver is currently doing, such as opening or closing a window or changing the radio station. We also expect that wrist worn devices are accurate enough to estimate the road quality. We use smartwatches since the adoption of smartwatches has been increasing over the past years, and is expected to keep increasing [9].

Figure 1.1: An example of a road with a bad road quality

Chapter 2

Research Questions

2.1 Summary of current work

In this research we follow up on our previous paper [8]. In this section we will summarize this paper and its conclusion to explain the reasons behind our research.

In [8] we made a distinction between 4 different fields of research into road safety. Below we include an excerpt of the publication that explains these fields.

> **Road maintenance** The research that focuses on road maintenance looks at the presence of irregularities in the road, such as potholes and bumps. This type of research tries to detect potholes and other issues that require maintenance on the road. This research usually tries to replace expensive machines that are now being used in road maintenance by detecting irregularities using vehicles which drive on the road anyway, such as buses, taxis, or regular traffic. Most research in this area uses a threshold based system to differentiate between bumps and normal irregularities.
>
> **Road pavement type monitoring** This type of research looks at the total roughness of certain stretches of road. Most research either use their own classification to classify a road, or use the International Roughness Index (IRI) equivalent.
>
> **Driving behavior** This research does not look at the roads but looks at the driving behaviour of people. this research either looks at the overall behaviour such as Junior et al. [10] and Guo et al. [11], or specifically at a certain aspect of driving, such as aggressive driver behaviour, for example the research of Jasinski et al. [12] or specifically at economical driving, such as Torp [13] and Magana et al. [14], Most of the driver behavior studies use data from the car, using OBD-II sensors to analyze the throttle input. Magana et al. [14] study how economic someone drives

by combining the OBD-II data, which includes both throttle application and engine status information, such as engine speed and engine load, with GPS and camera information.

All these research require adding an OBD-II sensor to your car. This has the benefit of getting a lot of data about the car, however it requires every user to add a component to their car. Also these research do not look at steering input.

Research into distracted driving, by J.Stutts et al. [15] [16], S. Klauer et al. [17] T. Ranney et al. [18], S. McEvoy [19], T. Dingus et al. [20], C. Pêcher [21], A. Stelling et al. [22] all indicate a higher crash risk and more unsafe driving behavior by performing distracting activities in the car, including rolling down a window, controlling the radio or car electronics, eating, drinking, making a phone call and having a conversation all increase the risk of being involved in a crash.

Research done by K. Young et al. [23] indicate that a high percentage of drivers, especially in the younger age ranges, are involved in these distracting activities while driving.

This leads us to believe that a way to monitor driving behavior that does not require acquiring additional measuring tools then already in possession by consumers can be a benefit to driver safety.

Safety The research into safety focuses on multiple aspect of safety. For example road design, what makes a road safe. Such as Schepers [7], who researched which design choices of a cycling path contributes to safety, or makes the safety worse. In this field most research done is on how to create safe roads, but not much research exists to automatically detect if a road/cycle path is safe.

Research done by A. Dijkstra [6] [3] [5] into road choice concludes that the route taken in a car highly influences the risk of an accident. One of the risks indicated in the paper is the difference in speeds at certain roads.

Table 9.1 at the end of the paper shows an overview of the existing research done in this area. In the previous paper we identified that all existing research is not yet capable of detecting driver behaviour. As shown in the table, P. Mohan et al. [24] developed partial behaviour tracking, but this is limited to detecting honking using the microphone of the smartphone. Previous research also has not covered the detection of unsafe situations on the road.

2.2 Research questions

As stated in the previous section, we identified two areas of research. The first is the detection of unsafe situations on roads. For example by detecting locations with tight turns or where heavy breaking occurs frequently. The second area covers the influence of driver behaviour on traffic safety.

Previous research mainly uses the movement sensors in smartphones, or dedicated sensors built into a car. Because of the fixed placement of these sensors it is not possible to detect the behaviour of the driver. We want to research if it is possible to detect the driver behaviour by using sensors attached to the driver. To make it more likely that drivers will be inclined to use this, we want to use consumer electronics that are already being used by consumers. We identified that due to the rise in smartwatch usage [9], this would be a viable wearable device to conduct this research.

To be able to research these items we created the following research questions:

1. Can we detect unsafe situations on the road using smartwatches?

2. Can we detect the behavior of a driver using a smartwatch?

To answer the first research question, we need to analyse the different factors impacting road safety. In our previous publication [8], we identified three factors that can identify road safety. Multiple research, such as done by A. Dijkstra [6] mention factors that civil engineers need to take into account in order to create safe roads. One of them is how tight a turn can be. Another way to determine whether a road is safe or not is to identify heavy breaking. Detection of heavy breaking alone is not sufficient, since we need to differentiate between heavy breaking due to structural deficiencies in the road design and heavy breaking due to other factors, such as driver inattention or mistakes by other road users. We estimate that we can differentiate between the two by looking at the amount of times drivers need to break heavily. The actual road quality is an other important factor for traffic safety. After all, poorly maintained roads can cause accidents and vehicle malfunctions.

Based on these considerations we divided the question up into the following subquestions:

1.1 Can we detect heavy breaking using smartwatch sensors?

1.2 Can we detect tight turns using smartwatch sensors?

1.3 Can we combine the data of multiple drives in the same road to differentiate between structural safety issues and occasional safety issues?

1.4 Can we classify the quality of the road?

It is important to detect whether a driver is doing something else besides driving, since distractions are one of the major reasons for traffic accidents. Several studies unveiled that talking on a cell phone increases the chance of an accident [25]. Other research indicates that different tasks, such as controlling the radio or portable music device, negatively affect safety [26] [27] [19] [17], Listening to music [28] [29] [21], eating or drinking are examples of such tasks [20] [15] [16]. Interestingly, it has been shown that a large percentage of drivers actively engage in distracting activities while participating in traffic, despite its known risks [23] [22] [30] [31] [18].

To be able to answer the second research question, we need to be able to detect what a driver is doing. We tried to identify actions that need input from the driver. Therefore we need to be able to detect what a driver does with their arms while behind the wheel. Since most people wear only one smartwatch, we noted that there would be a difference between actions with the arm with and without a smartwatch. We want to be able to discriminate a situation where a driver has their hands on the steering wheel from a situation where a driver is doing something else with their arm. We assumed that we could detect the actual action with the arm wearing the smartwatch, and that we could only detect whether the arm without the smartwatch is on the steering wheel or not.

We divided the second question: ”Can we detect the behavior of a driver using smartwatches?” in the following questions:

2.1 Can we detect if the arm wearing the smartwatch is on the steering wheel?

2.2 Can we detect which action the driver is performing with the arm wearing a smartwatch?

2.3 Can we detect if the driver is performing an action with the arm not wearing a smartwatch?

Figure 2.1: A distracted driver

Chapter 3

Requirements

The requirements are created using the MoSCoW method. We identify three stakeholders of the project.

The first one are the maintainers of the road. The road maintainers need a system to detect roads with structural issues, such as potholes and bad road quality. To a lesser extend this group also needs information about which road sections have a higher likelihood to cause unsafe situations. Currently, road maintenance requires specialized and expensive equipment, such as shown in figure 3.1

Figure 3.1: Example of a specialized vehicle used during maintenance and construction of the roads

The second stakeholder are policy makers. For this group the system needs to identify which roads cause more unsafe situations, so they can create measures to increase the safety of the road, either by replacing the dangerous situation or mitigate the dangers for example by lowering the speed limits. They also benefit from behaviour tracking. For example if unsafe situations occur more often when a driver is using specific features of the car, like changing the heat or is busy with the radio, they can either make legislation about it, or start awareness campaigns. Research into distracted driving, by J.Stutts et al. [15] [16], S. Klauer et al. [17] T. Ranney et al. [18], S. McEvoy [19], T. Dingus et al. [20], C Pêcher [21], A. Stelling et al. [22] indicate that distractions such as handling the car radio increase the change of accidents.

We also identify the user wearing the watch as the third stakeholder. The user could also benefit from some information given by the system. Although we recognize that this will be necessary to give incentives to actually use the system, this will be out of scope for the current research, and could be added in future work. We did include the functionality for the user in the list of requirements, but as won't have.

There are two main differences between road maintainers and policy makers in our definition. The first is that the road maintainers are interested in the quality of the road surface, so they know where repairs are neccesary. The policy makers have no specific interest in the quality of the road surface, but more about the safety of road, for example the gradient of the corners. The second is that policy makers also have an interest in information about the behaviour of the drivers.

3.1 Must have

Requirements	**Stakeholders**		
	Road maintainers	**Policy makers**	**Drivers**
Tracking of arm movement	X	X	X
Detecting movement of the car	X	X	X
Give information about the input the user has on the steering wheel	X	X	X
The system works for users driving a car	X	X	X
The system knows the location of the measurements	X	X	

3.2 Should have

Requirements	Stakeholders		
	Road maintainers	Policy makers	Drivers
The system is able to detect when a user takes one of their hands off the steering wheel.		X	
The system is able to detect anomalies in the road, such as potholes or bumps.	X	X	
The system gives information about the quality of the road.	X	X	
The system combines the data of multiple users to provide more accurate information about road quality.	X	X	X
The system will have minimal impact on battery life of the watch.			X
The system will have minimal impact on battery life on the phone if a phone is required to transmit data.			X

3.3 Could have

Requirements	Stakeholders		
	Road maintainers	Policy makers	Drivers
The system will detect whether a swerve is caused after the user takes their hand off the steering wheel.		X	
The system will process the data on the watch or phone.			X

3.4 Won't have

Requirements	**Stakeholders**		
	Road maintainers	**Policy makers**	**Drivers**
The system will process the data on the watch.			X
The system will give real time information to the user about road quality.			X
The system will give real time information to the user about road safety.			X
The system will give information to the user about their driving behaviour.			X
The system will give information how the user can improve their driving behaviour.			X
The system will give route suggestions based on the quality and safety of roads.			X
The system will use for other types of transport, such as bicycles, motor bikes and pedestrians.	X	X	X

Chapter 4

System architecture for data collection

For our system to work, we will need to collect the sensor data from the smartwatch, in order to analyse it later. The collection of the data is split between different areas. At first the location data and sensor data is gathered using a wrist worn device. Second this data will have to be sent to an external data storage to keep the memory requirements of the wearable as low as possible.

Next to this we need to be able to gather the ground truth. For the ground truth we need to be able to register what a driver was doing at a certain measurement, and where the driver was doing this.

As explained in the chapter 2 we have chosen a wrist worn device because this will give information of the movement of one of the arms of the driver. We decided to use a smartwatch since the adoption rate of smartwatches is rising [9], and we do not think people will buy a specific wrist worn sensing device just for measuring data.

figure 4.1 shows the most basic implementation of the sensor data generation. In this scenario the watch will directly send the measured data to a database.

Figure 4.1: The basic communication scheme

Since the most popular smartwatches do not have their own data connection, nor do they have enough data to store all the gathered data locally, we will need an intermediary device to collect the data. After receiving the data, the intermediary device can store it in a database, as shown in figure 4.2

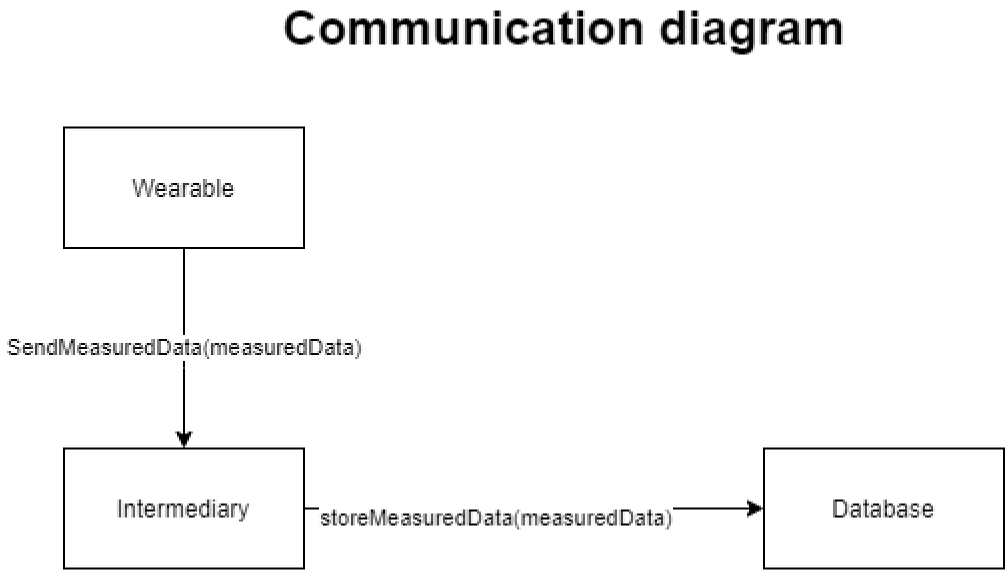

Figure 4.2: The basic communication scheme with an intermediary

In chapter 3 we identified the requirements for the end system.

For the requirements of Tracking of arm movement, Detecting movement of the car and Give information about the input the user has on the steering wheel the system will have to be able to track movements. For this we require that the system is equipped with movement sensors. To make accurate estimations of the arm movement, both the directional movement of the arm needs to be measured (x, y,z) as well as the rotational movement of the arm (pitch, power, roll). To make these measurements we decided to use the combination of an accelerometer for the directional movements, and a gyroscope for the rotational movements. To be able to tell on which roads the measurements of the driver were made, it is necessary to know the location of the driver. For this we chose to have a device with GPS. To make the measurements as accurate as possible it is beneficial to have a sample rate as high as possible. However, to minimize the impact on the battery life on both the intermediary and the watch, is is beneficial to have a very low sample rate. To be able to find a suitable sample rate that gives the best compromise between the two, we choose to have a system in which the sample rate can be modified. To be able to send the data to a database for processing and the data analysis, it is necessary to be able to have a connection to an intermediary device, such as a Bluetooth connection to a smartphone. To satisfy the requirements for energy usage, it is necessary to be able to drive for multiple hours without needing to charge the wearable device.

To be able to store the data, we will need to be able to create our own software for the system. The software of the system will have the same requirements as the system. The software on the wearable needs to record the data from the different sensors, (gyroscope, GPS and accelerometer), after which the data will need to be stored in the database. To be able to conduct our research we also need to collect the ground truth. To collect the ground truth it is necessary to store the activity that the driver is performing, as well as being able to sync it to the measured data. A communication diagram showing this interaction is shown in figure 4.3. We created an activity diagram in figure 4.4

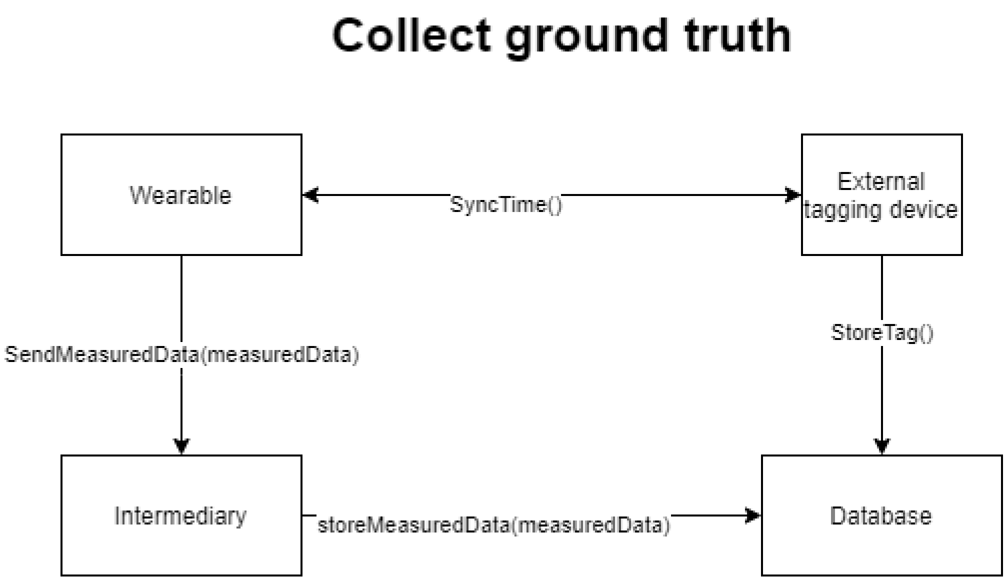

Figure 4.3: The basic communication scheme including ground truth

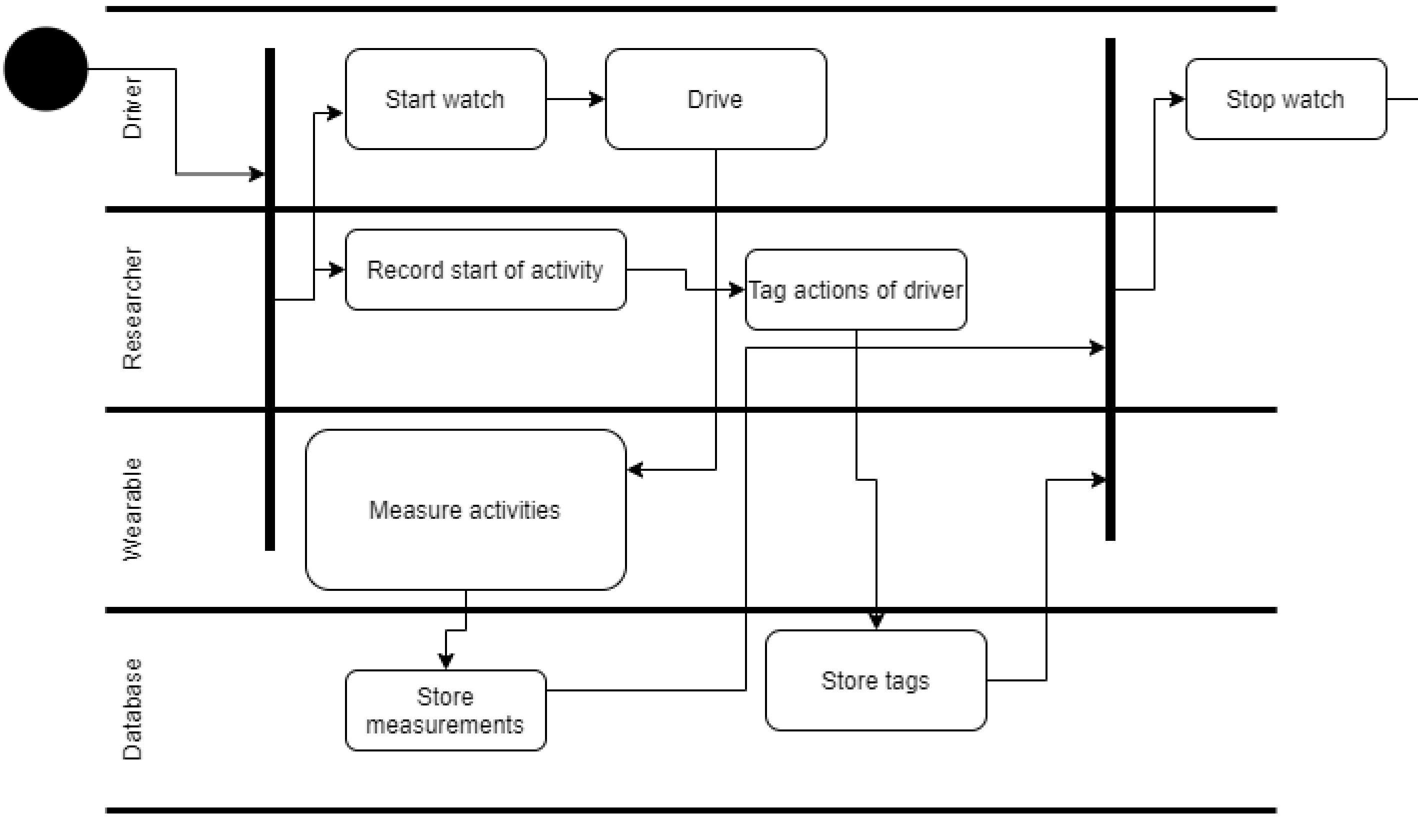

Figure 4.4: The activity diagram of the system

Based on the previous information we gathered the following list of requirements for the wearable device

- Accelerometer
- Gyroscope
- GPS
- Connection to an external database
- Multiple hours battery time
- Programmable

Chapter 5

Implementation of data collection

5.1 Devices

When we chose a watch we took into account several criteria, as stated in the previous chapter. Sensors: we required a watch with at least an accelerometer, gyroscope and GPS Software: We required a watch with an SDK or API that would allow us to read and store both the accelerometer, gyroscope and GPS data. Battery usage: Since we want to be able to drive without draining the battery of the smart watch, we wanted a watch that would be energy efficient.

After the first 2 criteria we were left with watches based on Android, Tizen (Samsung watches) and Garmin watches. Since the Garmin watches are more restricted in their functionality, as they are mainly aimed at sports enthusiasts, they also have a better battery life. Because of this reason we went for Garmin.

Phone OS: We decided to go for Android. Since Garmin has APIs both for android and IOS, we went for android since that did not require gaining a developer license.

5.2 The watch

We went for the most affordable watch in the Garmin arsenal that fit our requirements. This meant we went for the Garmin Vivoactive 4 to collect the data. The watch is equipped with an accelerometer and gyroscope to measure the movement data. It contains GPS for location data, with the possibility to combine it with either Galileo or Glonass.

To gather the sensor data it is required to subscribe to receive sensordata events. The SDK collects the data for a certain timeframe, after which it sends an event to the program, so it is possible to process the sensor data. It is possible to select 1, 2, 3 or 4 seconds as a timeframe. For example using a sample rate of 10 measurements per seconds, and a timeframe of 4 seconds, will generate 40 measurements every 4 seconds.

To gather location data it is also required to subscribe to receive location data. For location data it is not possible to choose the time interval, the watch will transmit a location every time it has a GPS-fix.

Figure 5.1: The smartwatch selected for this research project, the Garmin Vivoactive 4

5.2.1 Initial design and limitations of the Garmin connect SDK

In the original design the watch would send the location data and movement sensor data every time it would receive the data. However the Garmin SDK has a queue of 1 item. This would mean that if the program would send the data to the phone, and somewhat later the movement sensors would have data to be sent, the movements sensor data would get lost because the location data would still be sent. Considering the delays in the SDK combined with the delay of sending data using Bluetooth Low Energy this would cause a significant package loss.

We circumvented this issue by combining the location data and accelerometer data. By setting the period of the accelerometer data to 4 seconds, it increased the chance that at least one location data point is being sent in between the time of 2 accelerometer measurements. Every time an accelerometer measurement was being sent, it would be stored on the watch. The next time a location would be sent it combines the accelerometer data with the location data and sends it to the phone. In case the previous accelerometer data has not been sent when new data is ready, it will sent the accelerometer data without the GPS data. This can only happen when the GPS has no fix.

5.3 The phone

5.3.1 Android app

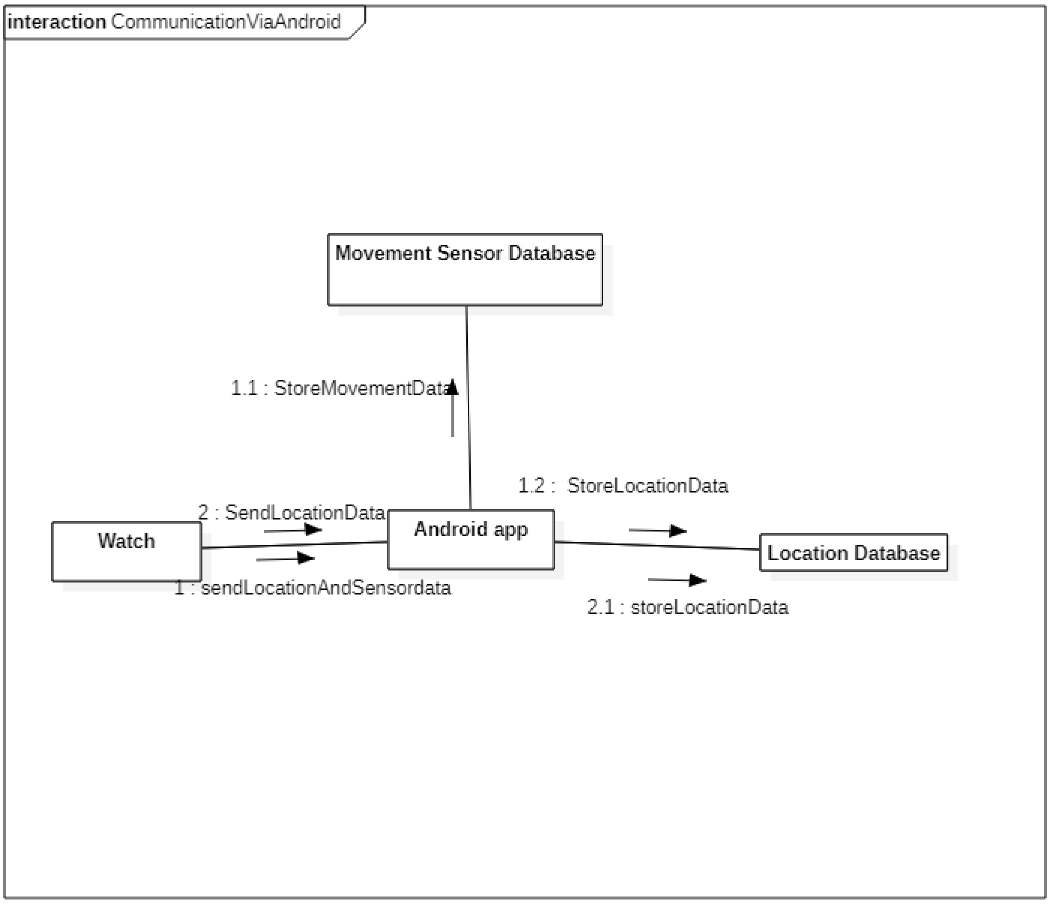

Figure 5.2: The simplified class diagram of the android application

We created an android app to collect the data being sent by the watch and store it in a sqlite database using the Room abstraction layer. As discussed in chapter 3 the software needs to be able to perform the following actions: Receive the measurements from the watch, and store it into a database. We used the library provided by the Garmin SDK to communicate between the watch and the phone. We made it a one way communication. The Garmin SDK gives an error on the watch when a message does not arrive at the app on the smartphone, so we do not need to send anything back to the watch. The app has 2 purposes. The first and most important purpose is to collect all the data from the watch and store it in a database. The second purpose is to give a passenger the possibility to tag in real time what the driver is doing. A diagram showing the communication between the watch and the app is shown in figure 5.3. We created a separate garminService, that can handle the communication of multiple watches, even though in our situation we only use one. This service is an android backgroundservice, meaning that it will go on even if the phone screen turns off or the user does something else with the phone. This service contains one, or theoretically multiple WatchConnectivity objects, which handle the actual communication to the smartwatch. When the watch sends back measurement data, the garminService broadcasts this to all WatchListeners. In our situation we had two watch listeners, one to show the progress on the screen, and one to send the measurements to the database, using the ResearchRun object. A simplified class diagram, hiding the android and front-end classes is shown in figure 5.2. An image of how the passenger is able to tag the actions of the driver is shown in figure 5.4

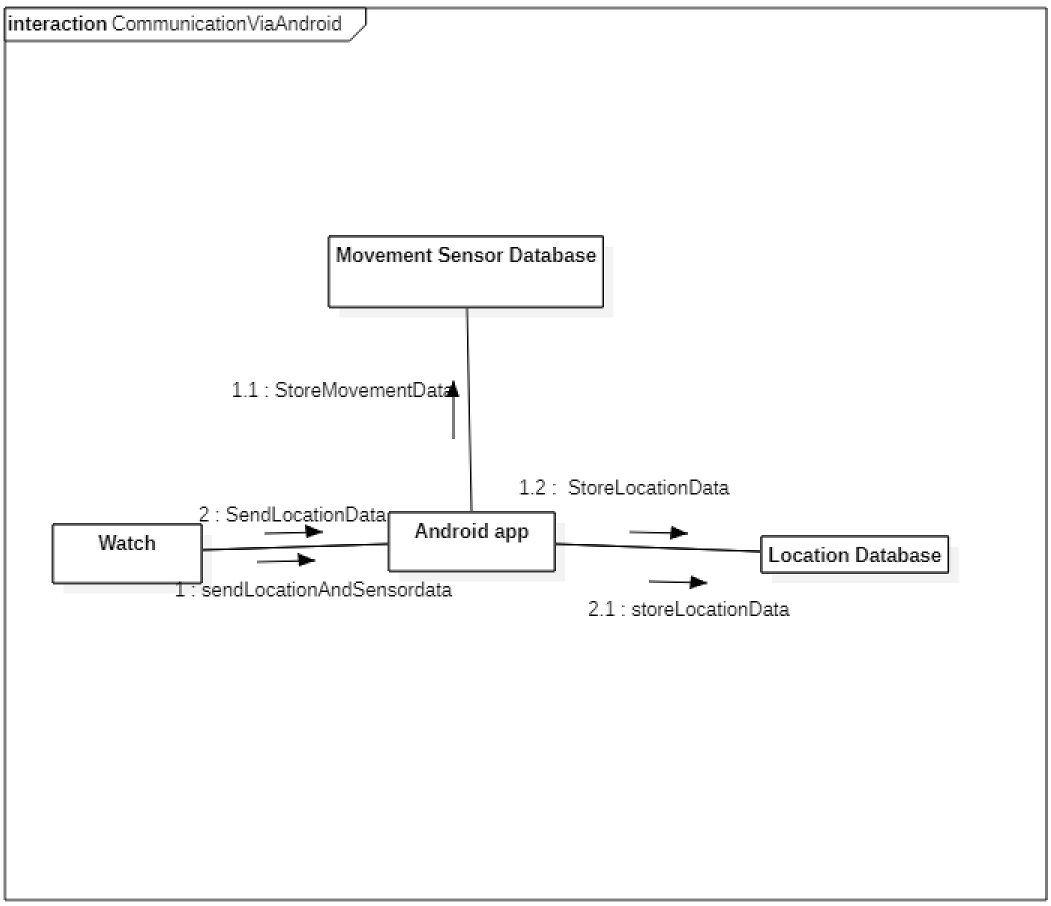

Figure 5.3: The original communication using an android app to collect the data

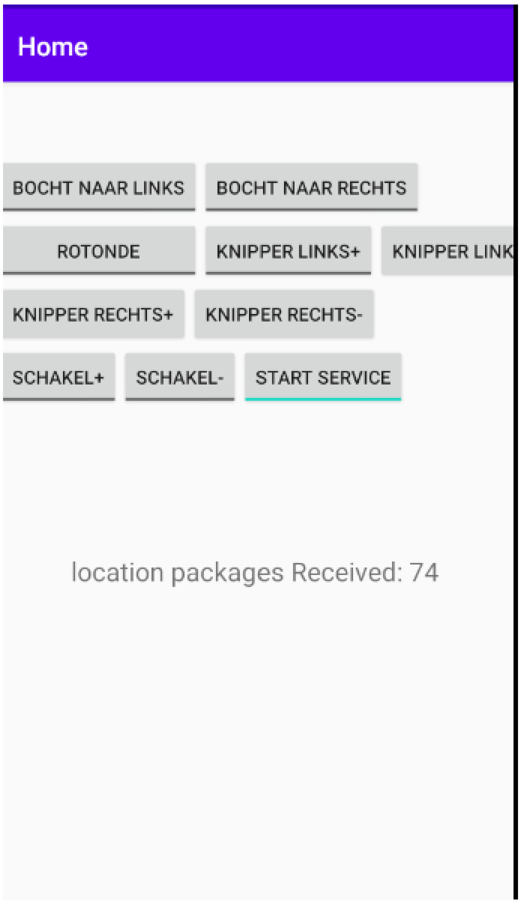

Figure 5.4: Example view of the android app

5.3.2 Limitations

During testing on the simulator this worked well after solving the issues that resulted from the limitations as discussed in section 5.2.1. However, during testing of the app using the smartwatch in connection with the phone, the communication between the watch and the phone was very unstable. Most of the messages would not come through, and after the first error the connection would not recover and every subsequent message would be lost. It would only work after restarting both the phone and the smartwatch. After sending some requests on the Garmin support forum, it was highly recommended to stop using the android library, and use the functionality to send data to a web server instead.

5.3.3 Web server

We created a small web application which receives the data from the smartwatch and stores it in a database. After that we redesigned the smartwatch app to send requests to this web server. Since the smartwatch does not have its own internet connection, it will send the data to the Garmin Connect app on the smartphone, which will make the web request. A diagram showing the communication from the watch to the web server is shown in figure 5.5

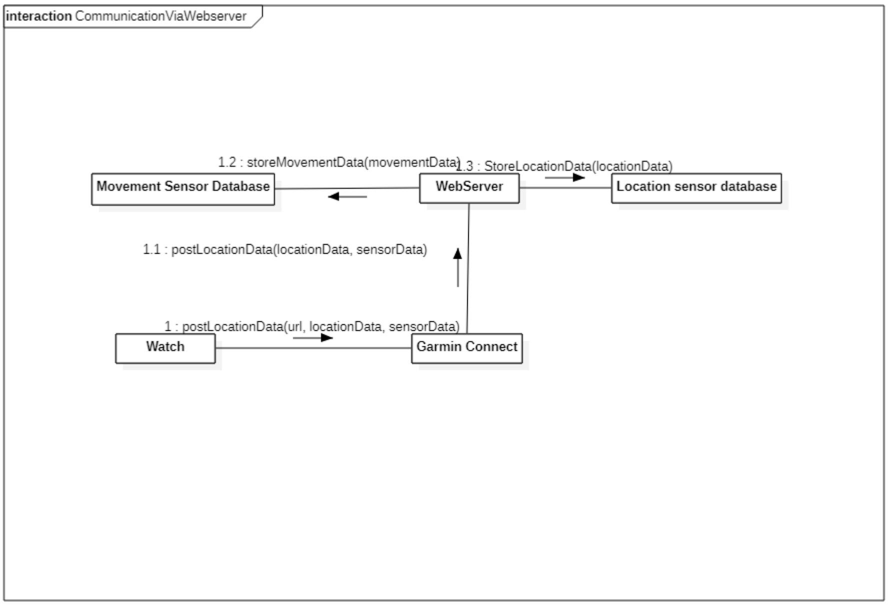

Figure 5.5: The new implementation using a web server to collect the data

Chapter 6

Data analysis

After gathering the data, our next step was to analyse it. In this chapter we will start by explaining how we gathered the ground truth, after which we will give an overview of existing methods and an explanation of which methods we used. After this we will explain how we did the implementation along with the results.

Figure 6.1: Gathering the ground truth

6.1 Ground truth

The original plan to tag the data was to use the app on the smartphone as discussed in section 5.3.1. In this situation a passenger would press the buttons when the driver did a certain action, such as turning a corner. However, due to the COVID-19 outbreak in the Netherlands during the period in which we were collecting our data, university regulations did not allow to bring a passenger in the car. Due to this limitation, we used the video camera on the smartphone to make a video of the driver. Using this video, we manually tagged all actions of the driver. We asked the driver to record the smartwatch showing the ID of the packages being sent, so it would be possible to synchronise the video to the data.

6.1.1 Limitations of ground truth due to the COVID-19 situation

The collection of the ground truth was heavily influenced by the evolving COVID-19 situation in the Netherlands. Due to the lockdown measures in place in the Netherlands, citizens were asked to keep a minimum distance of 1.5 meters from one another. Therefore it was also not allowed to be in a car with more than 1 person. Moreover, it was highly discouraged to go outside in non-vital situations. Because of this we needed to gather data by asking people that had a reason to drive somewhere. Because the persons would have to drive in their own car, it was also impractical to fit multiple cameras. In accordance to these limitations we found one person that gathered most of the data by driving to work. We also used a second person who drove to work, however because of the lockdown rules this was a very limited set.

Since it was impractical to fit the car with cameras, we used the cellphones of the drivers to record videos of the trips. We experimented both with using the front and back camera, and we found that if we used the front camera, it could give us information about the driver and the road the driver was using. However this meant that evaluating the road quality on the images was harder. This meant the amount of data we could gather about the road quality was limited.

In the end we came with about 9 hours of data, spread over 44 trips. Every 10 minutes of data cost about 1 to 1.5 hours of manual processing to gather the data

Amount of drivers	2
Amount of videos	44
Hours of video	9

6.2 Possible methods

In this section we will give a short overview of the possible machine learning techniques available to analyse our data, as well as explain how these methods will fit in for our data-set.

6.2.1 Support vector machine

A Support Vector Machine is a supervised learning model which is mainly used for classification. Several researches have approached the problems we are facing using SVMs. Bello-Salau et al. [32], Tai et al. [33], Seraj et al. [34], A. Fox et al. [35] and A. Allouch et al. [36] have used SVM's to measure the road quality. T. Brisimi et al. [37] used a Sparse SVM to gather the data. None of the aforementioned researches were tracking the behaviour of the driver.

6.2.2 SOM

Self Organising Maps, also known as Kohonen Maps, were invented in 1982 by T. Kohonen [38]. Self Organising Maps are originally meant for dimensionality reduction, but are nowadays mostly used for clustering. Some research has been done in using Self Organising Maps to measure the road quality, such as the research by Seraj et al. [39]. However this research only analyzes the road quality, not the behaviour.

6.2.3 Deep learning

Since one of our objectives is to detect driver behavior, we also looked into the field of Human Activity Recognition. Research into this area uses accelerometer and gyroscope data to classify certain behaviour of test subjects. One of the methods researchers have been using is deep learning. In the next sections we will discuss some of the deep learning techniques that have been used for human activity recognition or road quality detection.

Most research into Human Activity Recognition using Deep learning treat the problem as a Time Series Classification problem.

Because for Human Activity Recognition using movement sensors it is usually not only important what is happening at a certain time, but also what has happened before this, most deep learning techniques used for this purpose will use results to influence the decisions in the future, for example by using some form of feed forward network.

6.2.4 LSTM-RNN

One of the deep learning techniques used for time series classification is Long short-term memory Neural networks. An LSTM is a recursive neural network, invented by S. Hochreiter et al. [40]. Some of the applications for LSTMs that are relevant for this research are time series classification and anomaly detection, and human action recognition [41] [42] [43] [44].

One of the improvements for human activity recognition is using Bidirectional LSTMs, such as used by Zhao et al. [45]. The original LSTM only uses data that has been learned from the past to classify current behavior. A downside of this is that for human behavior context can be very important. One of the situations where future information might be important, is on a roundabout. Driving on a roundabout in countries which drive on the right side of the road such as the Netherlands means that you will enter the roundabout by taking a right corner, and exiting the roundabout will also involve taking a right corner. This means that it can be beneficial to know what happens in the future when trying to predict an action. A bidirectional LSTM uses two separate LSTMs, of which one will process the data from start to end, and the other LSTM will process the data from the end to the beginning, as shown in figure 6.2. By using a bidirectionally LSTM the accuracy of time series classification can be greatly improved, as shown by Zhao et al. [45]

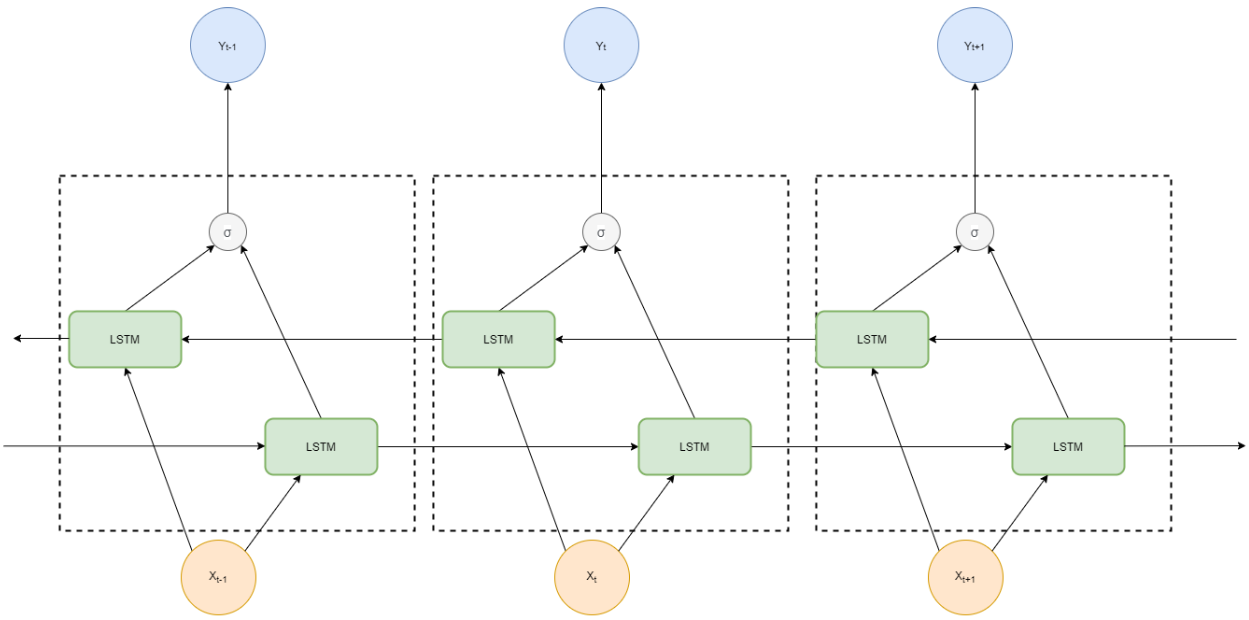

Figure 6.2: A Bidirectional LSTM

K. Saleh et al. [46] researched the use of Long short-term memory networks to classify driver behavior. In this case driver behavior does not mean what a driver is doing, but if the driving style of a driver is normal, aggressive, or drowsy. The research used the accelerometer and Gyroscope sensors in smart phones. This research used a Stacked-LSTM deep learning network to classify the driving style. The research showed promising results, with a much higher accuracy compared to Drivesafe [47]. This research shows that using LSTM Neural Networks are promising to classify driving behaviour.

6.2.5 HIVE-COTE

Another methodology that has been used for research into Time Series Classification is HIVE-COTE. HIVE-COTE is a modification to COTE. COTE was invented by Bagnall et al. [48]. HIVE-COTE was created by J. Lines et al. [49]. This was a huge improvement over COTE, and considered state of the art when it was created. However, as identified by I. Fawaz et al. [50], the accuracy comes at the cost of a computational performance. The high computational intensity means it is unviable for large data sets.

6.2.6 InceptionTime

I. Fawaz et al. [51] identified the potential of deep learning to create a machine learning tool with a high accuracy while still maintaining a Complexity low enough to make it feasible for large datasets. The result of this is InceptionTime. InceptionTime is a Time Series Classification implementation of the Inception-v4 architecture, as created by C. Szegedy et al. [52]. InceptionTime managed to perform at a high accuracy then HIVE-COTE, while performing much faster.

6.3 Implementation

6.3.1 Data Preprocessing

After gathering the data we had the labels as classified in figure 6.3. Since we had a limited arangement of cameras, we had issues always identifying when the driver was performing gear shifts. Sometimes because it was not clear if it was an up or downshift, but also sometimes because we were not sure if the drivers just put their hands on the arm rest or if they were actually shifting. Because of this we also added a classifier ”Right hand”. This was for all the situations where we were not sure what the driver was doing.

Label
Move a lane to the left
Adjusting the window
Bump
Drive roundabout
Enter/exit roundabout
fiddle with hair
Heavy Acceleration
Heavy Deceleration
Left blinker off
Left blinker on
Left corner
Left hand on knee
Left hand up
Look at watch
Move a lane to the right
Picking nose
Radio controls
Right blinker off
Right blinker on
Right corner
Right hand
Right hand mirror adjustment
Rough
Shift gear down
Shift gear up
Slight acceleration
Slight deceleration
slight left turn
slight right turn
Smooth
Sneezing
Wait for traffic light

Figure 6.3: The labels from data gathering

Most Human Activity Recognition projects assume that someone is doing one activity at a time. For example someone is either sitting, or standing, or running. However, for our purpose, we needed to detect certain events that occur on the same time. Some of the possible combinations that happen during driving are turning off the blinkers while turning a corner and shifting while turning a corner. Because of this we identified the need to split our categories in multiple classes. We identified 5 distinct classes. The classes with their activities are shown in figure 6.4

Car
The Car class contains all actions that have a direct impact on the direction of the car. All these actions involve turning the steering wheel, except waiting for the traffic light. In this case the car is not moving.
Left hand
The left hand class involves any action done with the left hand. Since our test subjects wore the smartwatch on the left hand, we expected the highest granularity in which actions could be identified. The actions include controlling the blinkers, but also different positions and actions of the left hand such as itching or the drivers fiddling with their hair.
Right hand
The right hand class involves any action done with the right hand. Since the watch was worn on the left hand with our experiment, we did not expect a high success rate in this class. However we anticipated that it might be possible to measure when the driver takes their right hand off the steering wheel to do something else.
Road surface
The Road surface class contains information about the road. Since we were not able to fit the car with more cameras, nor were we allowed to sit next to the driver to make notes about the road surface, as explained in section 6.1.1 our Road quality class is very basic. We made the distinction between a smooth surface, such as asphalt and a rough surface, such as cobblestone roads, or badly maintained asphalt. We also added speed bumps to the class, since they were possible to detect using our camera arrangement.
Acceleration
The acceleration class involves any action that involves the change of speed. We included this category since a hard deceleration might indicate that a dangerous situation has unfolded. We made the distinction between keeping the same speed, Heavy acceleration and deceleration, such as heavy breaking, and slight acceleration and deceleration, such as gentle breaking.

Car	Left hand	Right hand	Road surface	Acceleration
Move a lane to the left	fiddle with hair	Radio controls	Bump	Heavy Acceleration
Move a lane to the right	Look at watch	Right hand	Smooth	Heavy Deceleration
Left corner	Left blinker on	Right hand mirror adjustment	Rough	Slight acceleration
Right corner	Left blinker off	Shift gear down		Slight deceleration
Enter/exit roundabout	Right blinker on	Shift gear up		
slight left turn	Right blinker off			
slight right turn	Left hand on knee			
Drive roundabout	Left hand up			
Wait for traffic light	Picking nose			
	Sneezing			
	Adjusting the window			

Figure 6.4: The labels from data gathering after sorting them in categories

When we created the labels we anticipated that we would need to combine some of the labels. We anticipated that this would be necessary either because a certain action is not performed enough, or that it is not possible to make a distinction between certain actions. We anticipated that we might need to combine most of the Right hand actions, as well as some of the car movement actions. Since we added very specific actions for the Left hand, we also anticipated that those might need to be combined. As explained further in section 6.3.2 this was indeed the case. The new activities are shown in figure 6.5.

Car	Left hand	Right hand	Road surface	Acceleration
Left corner	Move hand to head	Right hand off steering wheel	Bump	Heavy Acceleration
Right corner	Look at watch		Smooth	Heavy Deceleration
Drive roundabout	Operate Blinkers		Rough	Slight acceleration
Wait for traffic light	Left hand not on steering wheel			Slight deceleration
	Adjusting the window			

Figure 6.5: The labels from data gathering after sorting them in categories

6.3.2 Machine learning

We shown some of the machine learning possibilities in 6.2. Since InceptionTime is at the moment the state of the art for Time Series Classification, this seems to be the most viable option. However there is a limitation to InceptionTime that makes it less ideal for our situation. Since InceptionTime is built for data-sets with only one activity at the time, it is not possible to use it with multiple output classes. To work around this problem we needed to train multiple InceptionTime models, one for each defined Class.

To test if training a single model with multiple outputs might be more suitable for our situation, we decided to also use a Bidirectional LSTM, as discussed in section 6.2.4.

Since many of the research into road quality measurements using smartphone sensors are using SVM, we also created a SVM model. We expected this to be mostly usable for the road quality measurements.

While analysing the data, we tried two approaches to deal with the part of the data that was simple straight driving. Since a large aspect of driving does not involve much actions, since most of the efficient routes contains mostly straight roads, with sometimes a roundabout or a corner, the data was largely biased towards straight driving. When analysing the different machine learning approaches, we took 2 approaches of dealing with the straight roads. The first approach is to include them, but work around it. For example changing the superparameters or the preprocessing to reduce the impact of the imbalanced data-set. Our second approach is to strip the Straight roads out of our data-set, and train a model on this data. Our assumption was that in this case the model would be able to more accurately make a distinction between the more specific activities.

SVM

When we started using the SVM model, we were mostly interested in the comparison for road quality. As we expected the model performed very poorly on the other categories. However, it also performed poorly at Road quality. Because we were limited to trips that were necessary, we were not able to select which roads the drivers should drive. This gave the problem that we mostly had smooth roads, and not so many bad quality roads. We did not manage to gain useful results

Acceleration The acceleration class is the one class with the most interesting results. As shown in figure 6.6, the model is not very accurate in predicting the acceleration. It does show some of the same trends as in the InceptionTime model. For example Most of the classes will mostly map to Slightly increasing.

	none	Tiny decrease	Tiny increase	small decrease	small increase
Fast decreasing	2.08%	0.00%	41.67%	56.25%	0.00%
Fast increasing	10.00%	0.00%	31.43%	58.57%	0.00%
decreasing	19.44%	0.00%	39.68%	40.88%	0.00%
increasing	11.66%	0.36%	44.23%	43.75%	0.00%
none	**45.72%**	0.71%	42.58%	10.84%	0.14%
Tiny decrease	14.08%	**0.93%**	78.53%	6.37%	0.09%
Tiny increase	10.87%	0.34%	**82.85%**	5.93%	0.00%
small decrease	19.61%	0.12%	61.76%	**18.52%**	0.00%
small increase	19.53%	0.29%	59.49%	20.56%	**0.12%**

Figure 6.6: Confusion Matrix for the acceleration class using a Support Vector Machine

InceptionTime

Because InceptionTime is the state of the art for Time Series Classification, we assumed this would lead to the best results. Since InceptionTime does not support multiple classes, we created separate models for each of the classes we defined. Most of the superparameters are already predefined or calculated based upon the data entered. However, we did experiment with combining the data. With this approach we take a collection of N measurements, and combine them into a batch. We then take the most occurring tag, and say that the entire set is the activity that is most prevalent in the batch. We found that it would not make a positive difference. In many cases it did not change anything, and in some cases the results became worse. Since there was no positive effect we did not include the results in the analysis.

Car

In our first approach we used all the data collected, including the straight roads. As shown in figure 6.7 this did not generate a useful model. The only occurrences where there were some predictions right was with taking a left turn, a right turn, or waiting for a traffic light. However even in these cases the accuracy was very low. We tried reducing the amount of Straight roads in steps to 10 percent, however this did not improve the results.

When we completely strip away the straight roads, the results get more interesting. While the accuracy of most of the tags is still not high enough, some trends are forming in this data. As shown in the Confusion matrix in figure 6.8 the model is not good at distinguishing right and left corners, as wel as right and left lane changes. An interesting trend is that it is much more prone to classify something as a Right turn than as a left turn. Our assumption is that this is because the route contains much more right turns then left turns. There is also a problem in distinguishing a lane change to the left from a lane change to the right.

	Driving	corner-left	corner-right	slight-left	Stop
Driving	99.69%	0.07%	0.19%	0.01%	0.05%
lane-left	100.00%	0.00%	0.00%	0.00%	0.00%
lane-right	100.00%	0.00%	0.00%	0.00%	0.00%
corner-left	94.15%	**4.11%**	1.74%	0.00%	0.00%
corner-right	94.23%	2.02%	**3.74%**	0.00%	0.02%
slight-left	99.90%	0.00%	0.00%	**0.10%**	0.00%
slight-right	99.89%	0.00%	0.00%	0.11%	0.00%
blinker-left-on	100.00%	0.00%	0.00%	0.00%	0.00%
blinker-left-off	100.00%	0.00%	0.00%	0.00%	0.00%
Stop	83.47%	0.00%	0.00%	0.00%	**16.53%**

Figure 6.7: Confusion Matrix for the Car class using InceptionTime

	lane-left	lane-right	corner-left	corner-right	slight-left	slight-right	blinker-left-off	stop
lane-left	**37.93%**	0.00%	40.23%	13.79%	0.00%	8.05%	0.00%	0.00%
lane-right	86.79%	**9.43%**	0.00%	0.00%	0.00%	3.77%	0.00%	0.00%
corner-left	0.00%	0.00%	**15.21%**	82.69%	1.72%	0.16%	0.00%	0.23%
corner-right	0.03%	0.00%	6.17%	**91.32%**	1.09%	0.47%	0.00%	0.92%
slight-left	0.20%	0.00%	0.10%	35.12%	**36.51%**	28.08%	0.00%	0.00%
slight-right	0.11%	0.00%	0.34%	33.52%	6.48%	**59.55%**	0.00%	0.00%
blinker-left-on	0.00%	0.00%	4.34%	40.46%	15.32%	39.88%	0.00%	0.00%
blinker-left-off	0.00%	0.00%	0.87%	0.00%	34.78%	56.52%	**7.83%**	0.00%
stop	0.00%	0.00%	0.00%	2.62%	0.00%	0.00%	0.00%	**97.38%**

Figure 6.8: Confusion Matrix for the Car class using InceptionTime Without straight roads

Left hand

When analysing the left hand class, we see the same trend as with the car class. when we include straight roads into the model, it will classify most activities as driving a straight road as shown in the confusion matrix in figure 6.9.

After removing the straight roads from the data-set, the role of Driving a straight roads get taken by itching, which in this case also includes fiddling with the drivers hair. An interesting relationship is between raising the left hand, and sneezing. This is probably because the driver sneezed into the left hand, which meant part of the movement was the same. The results are shown in the confusion matrix in 6.10

	Driving	Left-hand-loose
Driving	**99.99%**	0.01%
Look-at-watch	100.00%	0.00%
blinker-left-on	100.00%	0.00%
left-hand-up	100.00%	0.00%
left-hand-loose	83.43%	**16.57%**
nose-scratching	99.83%	0.17%
sneezing	100.00%	0.00%

Figure 6.9: Confusion Matrix for the left hand class using InceptionTime

	left-hand-up	left-hand-loose	nose-itching
look-at-watch	0.00%	2.70%	97.30%
blinker-left-on	0.00%	0.18%	99.82%
left-hand-up	**12.00%**	4.00%	84.00%
left-hand-loose	0.46%	**28.42%**	71.12%
nose-itching	0.09%	2.17%	**97.75%**
sneezing	13.71%	0.81%	85.48%

Figure 6.10: Confusion Matrix for the left hand class using InceptionTime without straight roads

Right hand The right hand is an interesting scenario. Since moving the right hand should not directly move the left hand, we did not expect it would be possible to see what a driver was doing with their right hand. We did expect it might be possible to detect when the driver was doing something with their right hand, because of corrections needed to be made with the left hand while the right hand is no longer holding the steering wheel. We tried learning the model while all our tags were still in place. After that did not work we tried to reduce the tags to controlling the radio and any other task. We chose these tags because the radio was one thing that we could clearly see in the video, as opposed to gear shifts. Also the driver needed to move forward to control the radio, while remaining in their chairs for the other right hand activities. This gave similar results to the previous approach. We shown the results in a confusion matrix in figure 6.11.

The next approach we took was combining all the activities that involved taking the right hand off the steering wheel into one. The results are shown in figure 6.12. While the results greatly improved as opposed to the previous approach, it is still not accurate enough. We did identify that increasing the amount of epochs improved the results.

	Driving	right-hand
Driving	99.93%	0.07%
radio	100.00%	0.00%
right-hand	98.72%	1.28%

Figure 6.11: Confusion Matrix for the Car class using InceptionTime

	Driving	right-hand
Driving	92.10%	7.90%
right-hand	46.40%	53.60%

Figure 6.12: Confusion Matrix for the Car class using InceptionTime Combining the radio and right hand

Road surface

As explained in section 6.1.1 the COVID-19 situation affected the amount of routes we could take, as well as the recording of the roads. Because of this we only classified 3 labels for the road surface monitoring. Smooth, bad and Speed bumps. We expected the model to have problems distinguishing speed bumps from bad road surface, as well as that most data would be classified as smooth, since the dataset was greatly unbalanced towards smooth roads. When analysing the data, of which the confusion matrix is shown if figure 6.13, we found that it had indeed some trouble identifying speed bumps from bad road surfaces, although it still has a 60 percent success rate. What did surprise us was the results for smooth and bad road surfaces. with percentages of 95 percent for bad road surfaces, and 99.77 percent chance that a smooth road surface was correctly identified. We conclude that using InceptionTime is a viable model to classify road surfaces.

	bump	rough	smooth
bump	**60.00%**	36.67%	3.33%
rough	0.00%	**95.45%**	4.55%
smooth	0.00%	0.23%	**99.77%**

Figure 6.13: Confusion Matrix for the Road quality class using InceptionTime

Acceleration

We divided both the acceleration and deceleration into 4 different labels, and added a label for no acceleration. The InceptionTime model does not manage to classify the acceleration and deceleration very sufficiently. It does not map anything to the larger acceleration en deceleration, and in the smaller changes it also have very low success rates, as shown in the confusion matrix in figure 6.14.

	none	Tiny decrease	Tiny increase	small decrease	small increase
Fast decreasing	2.08%	2.08%	33.33%	12.50%	50.00%
Fast increasing	8.57%	0.00%	31.43%	4.29%	55.71%
decreasing	20.64%	0.13%	31.50%	18.10%	29.62%
increasing	13.46%	0.48%	37.26%	13.22%	35.58%
none	**50.21%**	2.07%	33.45%	7.77%	6.49%
Tiny decrease	10.73%	**3.07%**	74.44%	6.83%	4.93%
Tiny increase	8.35%	2.03%	**79.25%**	6.28%	4.10%
small decrease	20.73%	1.16%	51.78%	**13.61%**	12.72%
small increase	19.98%	1.33%	50.66%	13.02%	**15.01%**

Figure 6.14: Confusion Matrix for the acceleration class using InceptionTime

LSTM

For the Long short-term memory neural network we used a bidirectional LSTM. Although it has been outperformed by InceptionTime for Time Series Classification, it has been successfully used for road surface classification. One of the advantages for our purpose is that it has the option of using multiple outputs, so it is possible to model our classes in one model. By incorporating the multiple classes into one model, we anticipated that it might improve the accuracy of the model. However, when analyzing we found the opposite. When we created a single model with multiple outputs, the accuracy dropped below 10 percent, and almost all labels would be classified as driving a straight road. Since this did not work we tried to approach this with the same tactic as the LSTM and SVM Approaches, by creating 5 seperate models. For The Car, Left hand, Right hand and Road quality models it performed poorly. We did get more promising results with the acceleration class

Acceleration

As with the InceptionTime model, it has issues with the larger accelerations and deceleration. However when the accelerations are less, the performance increases to over 80 percent. Although this model is still not good enough to predict the acceleration, it is better than the InceptionTime model as indicated by the confusion matrix in table 6.15.

	decreasing	increasing	none	Tiny decrease	Tiny increase	small decrease	small increase
Fast decreasing	0.00%	58.14%	0.00%	0.00%	0.00%	32.56%	9.30%
Fast increasing	0.00%	34.85%	4.55%	0.00%	0.00%	42.42%	18.18%
decreasing	**1.66%**	10.56%	3.32%	0.36%	0.12%	69.28%	14.71%
increasing	4.68%	**23.15%**	2.51%	0.34%	0.68%	33.64%	35.01%
none	0.17%	1.70%	**67.37%**	4.25%	5.35%	12.57%	8.58%
Tiny decrease	0.04%	0.34%	2.77%	**69.65%**	2.00%	21.46%	3.75%
Tiny increase	0.00%	0.15%	0.48%	0.51%	**84.10%**	3.91%	10.86%
small decrease	0.30%	1.89%	2.16%	3.79%	0.68%	**80.50%**	10.68%
small increase	1.51%	3.67%	1.37%	0.28%	16.63%	11.92%	**64.63%**

Figure 6.15: Confusion Matrix for the acceleration class using a Bidirectional LSTM

Acceleration

We found that measuring the acceleration using our machine learning models did not yield usable results. To try if it would still be possible to determine acceleration and deceleration, we decided to use the mathematical approach. The GPS data provided by the watch includes the current speed, so we decided to try to calculate the trend of the speed by performing linear regression. Using manual analysis we found that this would be a viable solution. The only problem using the GPS data is that it updates slower then the movement sensor data. This means it might take a second before detecting a quick brake. We recommend that this approach would be researched in future research.

Chapter 7

Evaluation

While performing the research, some circumstances outside of our control heavily influenced the data gathering. We think that the results would be more effective if it would have been possible to gather data from more persons, on different roads. The other limitation that we suspect highly influenced our results is the accuracy of the ground truth. We think that it is necessary to use more cameras to record both the actions of the driver as well as the circumstances on the road. Using different roads to gain a more diverse amount of road surfaces would also have been beneficial for this research.

In the introduction we stated the following research questions.

1. Can we detect unsafe situations on the road using smartwatches?
2. Can we detect the behavior of a driver using a smartwatch?

7.1 Can we detect unsafe situations on the road using smartwatches?

To analyse if we can detect unsafe situations on the road using smartwatches we will first analyse the subquestions:

1.1: Can we detect heavy breaking using smartwatch sensors?

We found out in section 6.3.2 that the models we used were not suitable to detect heavy breaking. However we did identify that it might be possible using the mathematical approach using GPS data instead of machine learning.

1.2: Can we detect tight turns using smartwatch sensors?

The analysis in section 6.3.2 shows that the models have problems differentiating between a slight turn and a larger turn.

1.3: Can we combine the data of multiple drives in the same road to differentiate between structural safety issues and occasional safety issues?

Since the models did not manage to accurately predict the safety issues, we did not try to combine the data of multiple drives.

1.4: Can we classify the quality of the road?

Using the InceptionTime model, we could successfully differentiate between bad road surfaces and smooth road surfaces. Due to the lack of different types of road we only have 2 labels saying something about the road quality. However, we assume that the current model could also be used for a more granular approach, when provided with enough data.

7.1.1 Conclusion

We did not manage to create a machine learning model that can accurately predict unsafe road situations. However, some of the limitations we ran into were caused by the local conditions due to the COVID-19 outbreak, so we think that it is viable to conduct further research into this area once the data gathering limitations are solved. We did manage to classify the roughness of the road based on the data we had.

7.2 Can we detect the behavior of a driver using a smartwatch?

To analyse if we can detect the behavior of a driver using smartwatches we start by answering the sub-questions.

2.1: Can we detect if the arm wearing the smartwatch is on the steering wheel?

It was not possible to detect whether the driver has their arm on the steering wheel. We suggest that this can be explained by the fact that we only recorded a few instances where the driver takes their left hand off the steering wheel. This is caused by the fact that most actions in the car that require you to take a hand off the steering wheel, are performed with the right hand.

2.2: Can we detect if the driver is performing an action with the arm not wearing a smartwatch?

Using InceptionTime it is possible to give a distinction between having the right hand on the steering wheel and doing something else. However, it was not possible to detect which activity the driver is performing.

2.3: Can we detect which action the driver is performing with the arm wearing a smartwatch?

It was not possible to detect which action the driver is performing with the arm wearing a smartwatch. We noticed that the amount of actions the driver performed while driving was quite low. We therefore expect that gathering more data could help in the creation of a working model.

7.2.1 Conclusion

Based on the available data in this project, we can not draw conclusions about a driver's behaviour or actions with their left hand. We did manage to detect whether a driver takes their right hand off the steering wheel.

7.3 Acknowledgements

I would like to thank ir. J. Scholten and dr.ir. B.J. van der Zwaag for the supervision and guidance during this research project, as well as helping me to get up to speed with the required techniques.

I would like to thank J. Klein Brinke, MSc for helping me with the machine learning techniques, as well as giving me feedback and giving me advice througout the process.

I would also like to thank everyone in the Pervasive Systems group for the insights and meaningful conversations I gathered during the period of my graduation project.

I would also like to thank prof. dr. ir. P.J.M. Havinga and ir. E. Molenkamp for being a part of my exam committee and reading my thesis.

I would also like my sister for helping me gather most of my research data, especially after my options of gaining data became more limited due to COVID-19.

Chapter 8

Future work

We identified certain areas that would be helped by further research. Firstly, we think that it is important to gather a more extensive data-set to train the machine learning models. Secondly, improving the InceptionTime model to support multiple output classes would be useful. We think this can be advantageous because this would allow the model to take certain aspects from one class into account during the analysis of other classes into account certain aspects from one class in the other. For example, turning on the blinkers will always be followed by either a corner or a lane change. Thirdly, we would also recommend to try to apply the HIVE-COTES model on the data-set using enough computing power. Since InceptionTime performs only slightly better then HIVES-COTE it will be useful to identify if HIVES-COTE may be more suitable for this data-set. One caveat is that HIVES-COTE requires much more computing power than InceptionTime. We would also recommend running InceptionTime with more Epochs. Due to limited computing power there was a maximum number of Epochs that was viable to run. Since for some of the data-sets, such as to classify whether the right hand is on the steering wheel, the performance was not converging yet, but we did not have enough computing power to continue this analysis.

Another aspect that would benefit from additional research is to first create a model of the human interaction in a car. If data can be gathered of which movements a driver makes while steering, controlling the radio and activities other than driving, a model could be made without the distortion of the car, the dampers, the movement etc. This might create more accurate models, although this does require the actual model to still take into account the movement of the car.

Chapter 9

Overview of Previous research

Table 9.1: Overview of existing technologies

paper	Vehicle	Sensors used			Devices used					behavior tracking	processing	technique used											
		accelerometer	gyroscope	gps	smartphone	tablet	car sensors	camera system	dedicated			SVM	SOM	SWT	Wavelet Decomposition	DWT	k means clustering	Roughness Index	Own/unknown	IRI	C4.5	Naive bayes	Sparse SVM
Bello-Salau et al. [32]	car	Y	Y	Y	N	N	Y	N	N	no	automated	Y	N	N	N	N	N	N	N	N	N	N	N
Tai et al. [33]	motorcycle	Y	Y	Y	Y	N	N	N	N	no	automated	Y	N	N	N	N	N	N	N	N	N	N	N
Seraj et al. [39]	wheelchair	Y	Y	Y	Y	N	N	N	N	no	automated	N	Y	N	N	N	N	N	N	N	N	N	N
Seraj et al. [53]	car	Y	Y	Y	Y	N	N	N	N	no	automated	N	N	Y	N	N	N	N	N	N	N	N	N
Seraj et al. [34]	car	Y	Y	Y	Y	N	N	N	Y	no	automated	Y	N	Y	N	N	N	N	N	N	N	N	N
Seraj et al. [54]	car, bike, train	Y	Y	Y	Y	N	N	N	N	no	automated	N	N	N	N	Y	N	N	N	N	N	N	N
Forslöf and Jones [55]	car, bike	Y	N	N	N	Y	N	N	N	no	automated	N	N	N	N	N	N	N	Y	N	N	N	N
Chih-Wei Yi et al. [56]	car	Y	Y	Y	Y	N	N	N	N	no	automated	N	N	N	N	N	N	N	Y	N	N	N	N
Kasun De Zoysa et al. [57]	Bus	Y	Y	Y	N	N	N	N	Y	no	automated	N	N	N	N	N	N	N	Y	N	N	N	N
P. Mohan et al. [24]	car	Y	Y	Y	Y	N	N	N	N	partial	automated	N	N	N	N	N	N	N	Y	N	N	N	N
J. Eriksson et al. [58]	car	Y	N	Y	N	N	N	N	Y	no	automated	N	N	N	N	N	N	N	Y	N	N	N	N
Dawkins et al. [59]	car	Y	Y	Y	N	N	N	N	Y	no	automated	N	N	N	N	N	N	N	Y	N	N	N	N
A. Mednis et al. [60]	car	Y	Y	Y	Y	N	N	N	N	no	automated	N	N	N	N	N	N	N	Y	N	N	N	N
L. Lima et al. [61]	car	Y	N	Y	Y	N	N	N	N	no	automated	N	N	N	N	N	N	N	Y	N	N	N	N
M. Ghadge et al. [62]	car	Y	N	N	Y	N	N	N	N	no	automated	N	N	N	N	N	Y	N	N	N	N	N	N
V. Douangphachanh et al. [63]	car	Y	Y	Y	Y	N	N	N	N	no	automated	N	N	N	N	N	N	Y	N	N	N	N	N
N. Silva [64]	car	Y	Y	Y	Y	N	N	N	N	no	automated	N	N	N	N	N	N	N	Y	N	N	N	N
A. Fox et al. [35]	car	Y	Y	Y	N	N	Y	N	N	no	automated	Y	N	N	N	N	N	N	N	N	N	N	N
X. Li and D. Goldberg [65]	car	Y	Y	Y	Y	N	N	N	N	no	automated	N	N	N	N	N	N	N	Y	N	N	N	N
P. Harikrishnan et al. [66]	car	Y	Y	Y	Y	N	N	N	N	no	automated	N	N	N	N	N	N	N	Y	N	N	N	N
T. Garbowski et al. [67]	unspecified	N	N	N	N	N	N	Y	N	no	automated	N	N	N	N	N	N	N	Y	N	N	N	N
K. Zang et al. [68]	bicycle	Y	Y	Y	Y	N	N	N	N	no	automated	N	N	N	N	N	N	N	N	Y	N	N	N
D. Rajamohan et al. [69]	car	Y	Y	Y	Y	N	N	Y	N	no	automated	N	N	N	N	N	N	N	N	Y	N	N	N
A. Allouch et al. [36]	car	Y	Y	Y	Y	N	N	N	N	no	automated	Y	N	N	N	N	N	N	N	N	Y	Y	N
T. Brisimi et al. [37]	car	Y	N	Y	Y	N	N	N	N	no	automated	N	N	N	N	N	N	N	Y	N	N	N	Y

Bibliography

[1] EuroRAP and Euro NCAP, “Roads that cars can read,” pp. 1–16, 2014. [Online]. Available: http://www.erf.be/images/RAINVISION/6._SS_Rainvision_March_2015.pdf

[2] Governors Highway Safety Association, “Everyone Walks. Understanding & addressing pedestrian safety,” pp. 1–73. [Online]. Available: http://www.ghsa.org/sites/default/files/2016-12/FINAL_sfped_0.pdf

[3] A. Dijkstra, “Testing the safety level of a road network ,” 2003. [Online]. Available: http://www.swov.nl/rapport/D-2003-15.pdf

[4] G. Wijlhuizen, A. Dijkstra, and J. v. Petegem, “Safe Cycling Network Ontwikkeling van een systeem ter beoordeling van de veiligheid van fietsinfrastructuur,” Tech. Rep., 2014. [Online]. Available: https://www.swov.nl/publicatie/safe-cycling-network

[5] A. Dijkstra, “Analyse van regionale verbindingen en routes,” no. Stichting Wetenschappelijk Onderzoek Verkeersveiligheid SWOV, 2010.

[6] ——, “En route to safer roads,” p. 247, 2011.

[7] J. P. Schepers, *A safer road environment for cyclists*, 2013.

[8] D. van der Zwet, “An overview of road safety research using consumerelectronics,” University of Twente, Tech. Rep., 2020.

[9] S. H. W. Chuah, P. A. Rauschnabel, N. Krey, B. Nguyen, T. Ramayah, and S. Lade, “Wearable technologies: The role of usefulness and visibility in smartwatch adoption,” *Computers in Human Behavior*, vol. 65, pp. 276–284, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.chb.2016.07.047

[10] J. F. Júnior, E. Carvalho, B. V. Ferreira, C. De Souza, Y. Suhara, A. Pentland, and G. Pessin, “Driver behavior profiling: An investigation with different smartphone sensors and machine learning,” *PLoS ONE*, vol. 12, no. 4, pp. 1–16, 2017.

[11] J. Guo, Y. Liu, L. Zhang, and Y. Wang, “Driving behaviour style study with a hybrid deep learning framework based on GPS data,” *Sustainability (Switzerland)*, vol. 10, no. 7, pp. 1–16, 2018.

[12] M. G. Jasinski and F. Baldo, “A Method to Identify Aggressive Driver Behaviour Based on Enriched GPS Data Analysis,” *GEOProcessing 2017 : The Ninth International Conference on Advanced Geographic Information Systems, Applications, and Services*, no. March 2017, pp. 97–102, 2017. [Online]. Available: https://www.thinkmind.org/download.php?articleid=geoprocessing_2017_6_20_38002

[13] K. Torp, “Evaluating Eco-driving Advice using GPS / CANBus Data,” pp. 44–53, 2013.

[14] V. C. Magana and M. Munoz-Organero, “Artemisa: A Personal Driving Assistant for Fuel Saving,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2437–2451, 2016.

[15] J. S. Stutts, E. A. Rodgman, L. Staplin, and I. Researcher, “Distractions in Everyday Driving AAA Foundation for Traffic Safety,” no. January, 2003.

[16] J. Stutts, J. Feaganes, D. Reinfurt, E. Rodgman, C. Hamlett, K. Gish, and L. Staplin, “Driver’s exposure to distractions in their natural driving environment,” *Accident Analysis and Prevention*, vol. 37, no. 6, pp. 1093–1101, 2005.

[17] S. Klauer, S. Klauer, T. a. Dingus, T. a. Dingus, V. L. Neale, V. L. Neale, J. Sudweeks, J. Sudweeks, D. Ramsey, and D. Ramsey, “The Impact of Driver Inattention On Near Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data,” *Analysis*, no. April, p. 226, 2006.

[18] T. Ranney, E. Mazzae, R. Garrott, and M. Goodman, “NHTSA driver distraction research: Past, present, and future,” *USDOT, National Highway Traffic Safety Administration*, pp. 1–11, 2000. [Online]. Available: http://www-nrd.nhtsa.dot.gov/departments/HumanFactors/driver-distraction/PDF/233.PDF

[19] S. P. McEvoy, M. R. Stevenson, and M. Woodward, “The impact of driver distraction on road safety: Results from a representative survey in two Australian states,” *Injury Prevention*, vol. 12, no. 4, pp. 242–247, 2006.

[20] T. A. Dingus, F. Guo, S. Lee, J. F. Antin, M. Perez, M. Buchanan-King, and J. Hankey, “Driver crash risk factors and prevalence evaluation using naturalistic driving data,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 10, pp. 2636–2641, 2016. [Online]. Available: http://www.pnas.org/lookup/doi/10.1073/pnas.1513271113

[21] C. Pêcher, C. Lemercier, and J. M. Cellier, "Emotions drive attention: Effects on driver's behaviour," *Safety Science*, vol. 47, no. 9, pp. 1254–1259, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.ssci.2009.03.011

[22] A. Stelling and M. P. Hagenzieker, "Afleiding in het verkeer: Een overzicht van de literatuur," *Leidschendam, Pays-Bas: SWOV*, 2012. [Online]. Available: http://www.pnas.org/lookup/doi/10.1073/pnas.1513271113

[23] K. L. Young and M. G. Lenné, "Driver engagement in distracting activities and the strategies used to minimise risk," *Safety Science*, vol. 48, no. 3, pp. 326–332, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.ssci.2009.10.008

[24] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones," p. 323, 2008.

[25] M. B. Neider, J. S. McCarley, J. A. Crowell, H. Kaczmarski, and A. F. Kramer, "Pedestrians, vehicles, and cell phones," *Accident Analysis and Prevention*, vol. 42, no. 2, pp. 589–594, 2010.

[26] K. Hurts, L. S. Angell, and M. A. Perez, "The Distracted Driver: Mechanisms, Models, and Measurement," *Reviews of Human Factors and Ergonomics*, vol. 7, no. 1, pp. 3–57, 2011.

[27] D. D. Salvucci, D. Markley, M. Zuber, and D. D. P. Brumby, "iPod Distraction : Effects of Portable Music-Player Use on Driver Performance," *CHI 2007 Proceedings: Task & Attention*, pp. 243–250, 2007. [Online]. Available: http://discovery.ucl.ac.uk/125955/

[28] D. De Waard, K. Edlinger, and K. Brookhuis, "Effects of listening to music, and of using a handheld and handsfree telephone on cycling behaviour," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 14, no. 6, pp. 626–637, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.trf.2011.07.001

[29] D. C. Schwebel, D. Stavrinos, K. W. Byington, T. Davis, E. Elizabeth, O. Neal, and D. D. Jong, "Distraction and Pedestrian Safety: How Talking on the Phone, Texting, and Listening to Music Impact Crossing the Street," vol. 45, no. 2, pp. 266–271, 2013.

[30] C. Goldenbeld, M. Houtenbos, E. Ehlers, and D. De Waard, "The use and risk of portable electronic devices while cycling among different age groups," *Journal of Safety Research*, vol. 43, no. 1, pp. 1–8, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jsr.2011.08.007

[31] L. L. Thompson, F. P. Rivara, R. C. Ayyagari, and B. E. Ebel, “Impact of social and technological distraction on pedestrian crossing behaviour: An observational study,” *Injury Prevention*, vol. 19, no. 4, pp. 232–237, 2013.

[32] H. Bello-Salau, A. Aibinu, A. Onumanyi, E. Onwuka, J. Dukiya, and H. Ohize, “New road anomaly detection and characterization algorithm for autonomous vehicles,” *Applied Computing and Informatics*, 2018. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S2210832718300309

[33] Y.-c. Tai, C.-w. Chan, and J. Y.-j. Hsu, “Automatic road anomaly detection using smart mobile device,” pp. 1–8, 2010. [Online]. Available: http://w.csie.org/~yctai/papers/taai2010_paper.pdf

[34] F. Seraj, B. v. d. Zwaag, and A. Dilo, “RoADS: A road pavement monitoring system for anomaly detection using smart phones,” *. . . Workshop on Modeling . . .*, 2014. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-29009-6_7

[35] A. Fox, B. V. Kumar, J. Chen, and F. Bai, “Multi-Lane Pothole Detection from Crowdsourced Undersampled Vehicle Sensor Data,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 12, pp. 3417–3430, 2017.

[36] A. Allouch, A. Koubaa, T. Abbes, and A. Ammar, “RoadSense: Smartphone Application to Estimate Road Conditions Using Accelerometer and Gyroscope,” *IEEE Sensors Journal*, vol. 17, no. 13, pp. 4231–4238, 2017.

[37] T. S. Brisimi, S. Ariafar, Y. Zhang, C. G. Cassandras, and I. C. Paschalidis, “Sensing and classifying roadway obstacles: The street bump anomaly detection and decision support system,” *IEEE International Conference on Automation Science and Engineering*, vol. 2015-Octob, pp. 1288–1293, 2015.

[38] T. Kohonen, “Self-organized formation of topologically correct feature maps,” *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[39] F. Seraj, P. J. Havinga, and N. Meratnia, “SpinSafe: An unsupervised smartphone-based wheelchair path monitoring system,” *2016 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops 2016*, no. PerMoby, 2016.

[40] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[41] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, “Sequential Deep Learning for Human Action Recognition,” in *Lecture Notes in Computer*

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2011, vol. 5477, pp. 29–39. [Online]. Available: http://link.springer.com/10.1007/978-3-642-25446-8_4

[42] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long Short Term Memory Networks for Anomaly Detection in Time Series," *ESANN 2012 proceedings, 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, no. April, pp. 1–650, 2012.

[43] J. Schmidhuber, D. Wierstra, and F. Gomez, "Evolino: Hybrid neuroevolution / optimal linear search for sequence learning," *IJCAI International Joint Conference on Artificial Intelligence*, pp. 853–858, 2005.

[44] G. P. Zhang, "Recurrent Neural Networks for Time Series Forecasting," *Handbook of Natural Computing*, vol. 1-4, pp. 461–477, 2012.

[45] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang, "Deep Residual Bidir-LSTM for Human Activity Recognition Using Wearable Sensors," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[46] K. Saleh, M. Hossny, and S. Nahavandi, "Driving behavior classification based on sensor data fusion using LSTM recurrent neural networks," pp. 1–6, 2018.

[47] E. Romera, L. M. Bergasa, and R. Arroyo, "Need data for driver behaviour analysis? Presenting the public UAH-DriveSet," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 387–392, 2016.

[48] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2522–2535, 2015.

[49] J. Lines, S. Taylor, and A. Bagnall, "HIVE-COTE: The Hierarchical Vote Collective of Transformation-based Ensembles for Time Series Classification," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 5, pp. 1–35, 2018.

[50] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[51] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "InceptionTime: Finding AlexNet for Time Series Classification," 2019. [Online]. Available: http://arxiv.org/abs/1909.04939

[52] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 4278–4284, 2017.

[53] F. Seraj, K. Zhang, O. Turkes, N. Meratnia, and P. J. M. Havinga, "A smartphone based method to enhance road pavement anomaly detection by analyzing the driver behavior," *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers - UbiComp '15*, pp. 1169–1177, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2800835.2800981

[54] F. Seraj, N. Meratnia, and P. J. Havinga, "RoVi: Continuous transport infrastructure monitoring framework for preventive maintenance," *2017 IEEE International Conference on Pervasive Computing and Communications, PerCom 2017*, pp. 217–226, 2017.

[55] Lars Forslöf and Hans Jones, "Roadroid: Continuous Road Condition Monitoring with Smart Phones," *Journal of Civil Engineering and Architecture*, vol. 9, no. 4, pp. 485–496, 2015. [Online]. Available: http://www.davidpublisher.org/index.php/Home/Article/index?id=7647.html

[56] C. W. Yi, Y. T. Chuang, and C. S. Nian, "Toward Crowdsourcing-Based Road Pavement Monitoring by Mobile Sensing Technologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1905–1917, 2015.

[57] K. De Zoysa, C. Keppitiyagama, G. P. Seneviratne, and W. W. A. T. Shihan, "A public transport system based sensor network for road surface condition monitoring," p. 1, 2008.

[58] B. H. Jakob Eriksson, Lewis Girod and H. B. Ryan Newton, Samuel Madden, "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring," pp. 515–525, 2012.

[59] J. Dawkins, R. Bishop, B. Powell, and D. Bevly, *Investigation of Pavement Maintenance Applications of Intellidrive - Implementation and Deployment Factors for Vehicle Probe-Based Pavement Maintenance*, 2011. [Online]. Available: http://cts.virginia.edu/PFS_PMSA99_Final.pdf

[60] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using Android smartphones with accelerometers," *2011 International Conference on Distributed Computing in Sensor Systems and Workshops, DCOSS'11*, no. 10, 2011.

[61] L. C. Lima, V. J. P. Amorim, I. M. Pereira, F. N. Ribeiro, and R. A. R. Oliveira, "Using crowdsourcing techniques and mobile devices for asphaltic pavement quality recognition," *Brazilian Symposium on Computing System Engineering, SBESC*, pp. 144–149, 2017.

[62] M. Ghadge, D. Pandey, and D. Kalbande, "Machine learning approach for predicting bumps on road," *Proceedings of the 2015 International Conference on Applied and Theoretical Computing and Communication Technology, iCATccT 2015*, pp. 481–485, 2016.

[63] V. Douangphachanh and H. Oneyama, "SmartRoadSense: Collaborative Road Surface Condition Monitoring," *The Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, no. c, pp. 433–439, 2013.

[64] N. Silva, J. Soares, V. Shah, M. Y. Santos, and H. Rodrigues, "Anomaly Detection in Roads with a Data Mining Approach," *Procedia Computer Science*, vol. 121, pp. 415–422, 2017. [Online]. Available: https://doi.org/10.1016/j.procs.2017.11.056

[65] X. Li and D. W. Goldberg, "Toward a mobile crowdsensing system for road surface assessment," *Computers, Environment and Urban Systems*, vol. 69, no. January, pp. 51–62, 2018. [Online]. Available: https://doi.org/10.1016/j.compenvurbsys.2017.12.005

[66] P. M. Harikrishnan and V. P. Gopi, "Vehicle Vibration Signal Processing for Road Surface Monitoring," *IEEE Sensors Journal*, vol. 17, no. 16, pp. 5192–5197, 2017.

[67] T. Garbowski and T. Gajewski, "Semi-automatic Inspection Tool of Pavement Condition from Three-dimensional Profile Scans," *Procedia Engineering*, vol. 172, pp. 310–318, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.proeng.2017.02.004

[68] K. Zang, J. Shen, H. Huang, M. Wan, and J. Shi, "Assessing and mapping of road surface roughness based on GPS and accelerometer sensors on bicycle-mounted smartphones," *Sensors (Switzerland)*, vol. 18, no. 3, 2018.

[69] D. Rajamohan, B. Gannu, and K. Rajan, "MAARGHA: A Prototype System for Road Condition and Surface Type Estimation by Fusing Multi-Sensor Data," *ISPRS International Journal of Geo-Information*, vol. 4, no. 3, pp. 1225–1245, 2015.