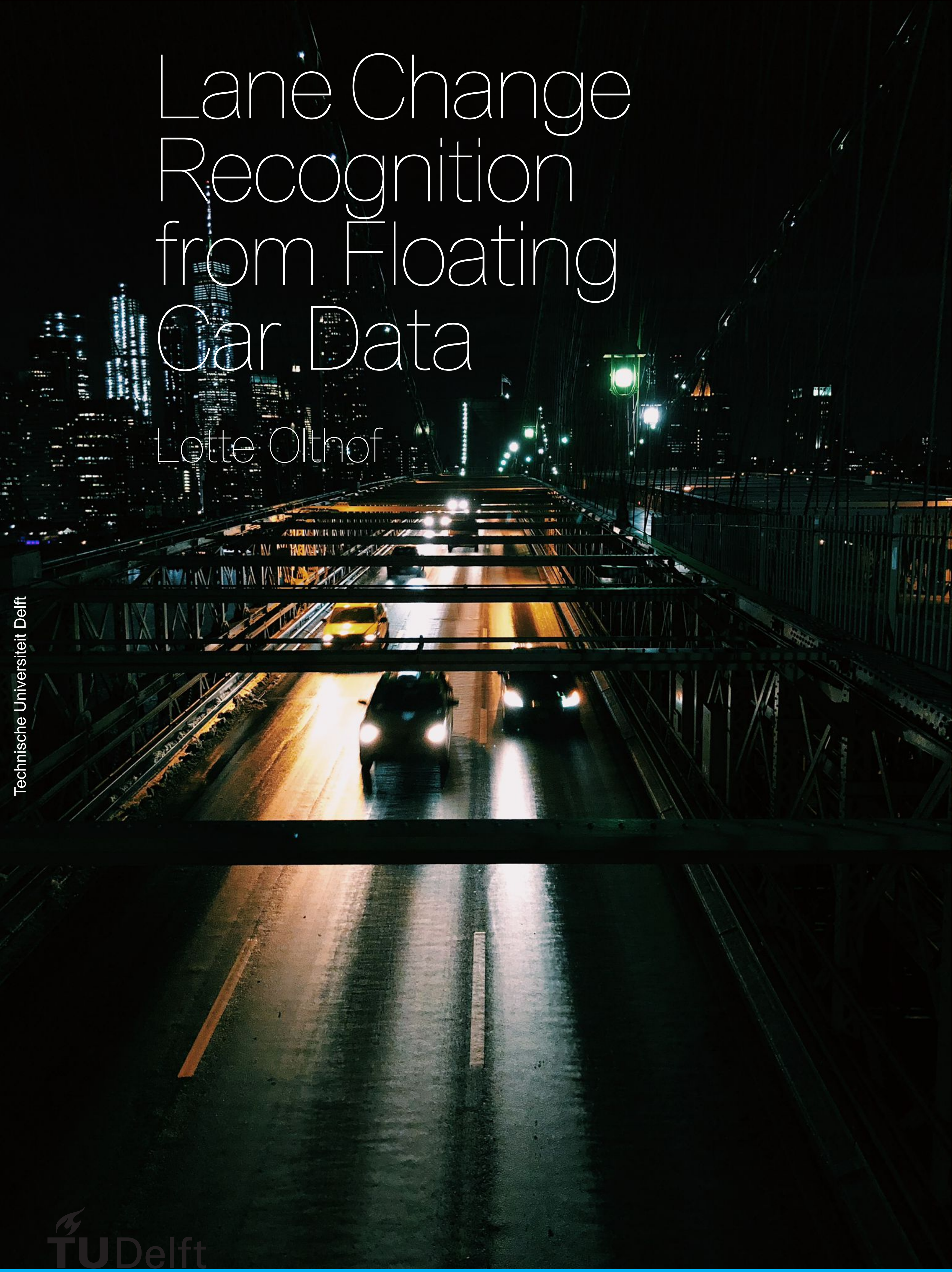
Lane Change Recognition from Floating Car Data

Lotte Olthof



Technische Universiteit Delft

Lane change recognition from Floating Car Data

by

Lotte Olthof

to obtain the degree of Master of Science

in Transport, Infrastructure & Logistics

at the Delft University of Technology,

to be defended on April 7, 2022 at 15:00.

Student number:	4725832	
Project duration:	April 1, 2021 – March 24, 2022	
Thesis committee:	Prof. dr. B. van Arem,	TU Delft, chair
	Dr.V.L. Knoop,	TU Delft, supervisor
	Dr. ir. J. C. F. de Winter,	TU Delft, supervisor
	Danielle Petit,	De Verkeersonderneming
	Jeroen Rijsdijk,	De Verkeersonderneming

Cover image by Sarah Hoogenboom

TU Delft

De Verkeersonderneming

Preface

Dear reader,

In front of you lies the result of many months of work, and the final result of my master thesis and study career. Writing this thesis has been an educational experience, both on a scientific and personal level. It has been the first time in my life working on a single project for this amount of time, and the first time feeling responsible for the result in such an extensive manner. Conducting this research for a large part during the Covid-19 pandemic added an extra challenge of not being able to meet and work together with many of my committee members and colleagues in real-life. Everyone I met and worked with on the way has however put a lot of effort in being available and offering opportunities to get in contact as much as possible, which I am very thankful for.

I would thereby also thoroughly like to thank all my TU Delft supervisors for their help, guidance, availability and support during my thesis period, and for making this project possible. A very special thanks to Dr. Victor Knoop for guiding me throughout the process with regular meetings, support, and feedback. I always experienced our meetings as very helpful and pleasant, and your support always offered me new motivation and optimism in continuing the research. Also a big thanks to Dr. ir. Joost de Winter for your comments and positive input, showing me the value of the research when I sometimes lost track of it. Finally, thanks a lot to Prof. dr. Bart van Arem for chairing my thesis committee and for your trust, feedback and approachability.

Furthermore I would also like to express my gratitude to De Verkeersonderneming for giving me the opportunity to conduct my research, providing me with help and support, and supplying the data for this research. A special thanks to Jeroen Rijsdijk for offering support and feedback, and for sharing your knowledge and advice, both content- and process-wise. Also a big thank you to Danielle Petit for guiding me through the process and giving me a push once in a while when needed. Without your encouragement and assistance this research would not have been the same. Lastly, I want to express my gratitude to the rest of the data team at De Verkeersonderneming for offering help and advice throughout the project, and giving me an insight in your projects.

A special thanks goes out to Mohammad Ali Arman for all the time and effort spent on the explanation and implementation of the trajectory reconstruction algorithm, which holds a crucial part in this research, and without which this thesis would not have been possible. I am very grateful for all your time and help.

Last but definitely not least, I want to thank my family, friends, and colleagues without who my thesis and entire study career would not have been the same. Thank you for always supporting me, checking up on me, and for making these years the shaping and enjoyable ones they have been. Especially to my family, who provided me the opportunity to follow this education, and who have always supported me in every possible way, for which I am very grateful.

Lotte Olthof
Delft, March 2022

Contents

List of Figures

List of Tables

Summary

With the increased use of in-car navigation devices and mobile phone navigation applications, large amounts of Floating Car Data are becoming available. This data source offers GPS traces and speed information of vehicles, and allows for traffic analysis. The extensive use of such navigation applications and systems also provides opportunities for (personalised) in-car advises and traffic management such as lane change advises, through which traffic flow can be improved. An issue currently withholding such advises however, is the inaccuracy of regular GPS, which can be offset up to several meters. It is therefore not possible to say with certainty in which lane a vehicle is driving, making it difficult to offer lane-level driving advises. By determining when a vehicle makes a lane change, a first step towards lane-level knowledge is gained. Furthermore, knowing if, and where a vehicle makes a lane change is also crucial to know whether a vehicle follows an advice given, and whether an advice would be beneficial. Currently no method exists which can, from Floating Car Data alone, with certainty deduct whether a vehicle makes a lane change, or whether the lateral movement seen in the data is caused by GPS error or distortion.
The goal of this research is to apply a new method for lane change recognition, and explore to what extent that method is able to deduct lane change manoeuvres from Floating Car Data, by looking into the following main research question:

”To what extent can lane changes be recognised from Floating Car Data?”.

In order to answer this question, a combination of Floating Car Data from Flitsmeister, and loop detector data, both from the Dutch highway A27 between Utrecht Noord and knp Eemnes, is used. First a data analysis is executed to examine the level at which information can reliably be deducted from Floating Car Data, after which a trajectory reconstruction algorithm by Arman and Tampere (2021) is applied in order to identify lane changes. By matching Floating Car Data trajectories with loop detector passages, vehicles are located on a specific lane at each loop detector location. From this lane changes being made in the road sections in-between these loop detector locations are deduced and labels indicating the type of lane change can be created. This is depicted in figure 1.

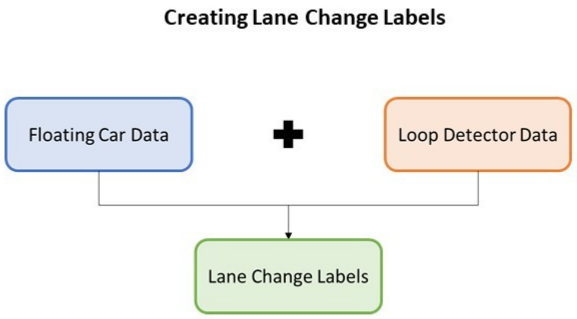


Figure 1: Schematic representation of data usage step 1

The lane change labels acquired through the trajectory reconstruction algorithm are then used for two methods of lane change recognition; a rule-based method, and a machine learning algorithm.

The first method for lane change recognition applied is a rule-based method by Van Ballegooijen (2019), which considers a vehicle's delta heading (Δh, in degrees, the difference in vehicle heading compared to the direction of the infrastructure), during several consecutive time steps of 1 second. A lane change is thereby defined according to the following definition, in which t_i represents time-step i, and Δh_{t_i} the delta heading of the vehicle at time-step i:

$$(\Delta h)\ at\ t_0,\ t_1,\ and\ t_2 < 0\ or\ (\Delta h)\ at\ t_0,\ t_1,\ and\ t_2 > 0$$

$$and$$

$$abs(\Delta h_{t_0} + \Delta h_{t_1} + \Delta h_{t_2} + \Delta h_{t_3}) >= 6$$

The resulting lane changes according to this definition are compared to those found through the trajectory reconstruction algorithm of Arman and Tampere (2021), as well as to the number expected according to literature. A schematic representation of this can be seen in figure 2.

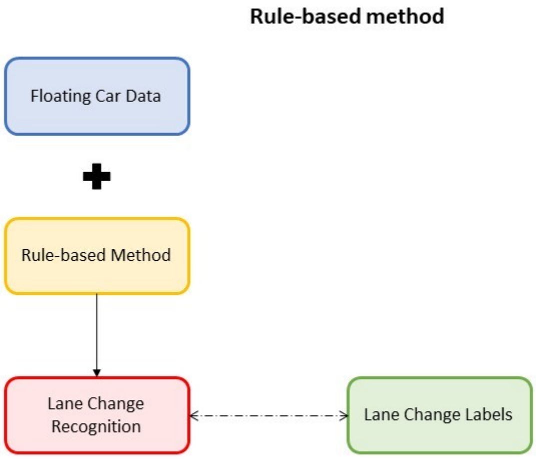


Figure 2: Schematic representation of rule based method

The second lane change recognition method applied is a Random Forest algorithm, which uses several different features for lane change recognition. Four different models are trained according to different types of lane change categorisations, namely:

- Lane Change Yes/No
- Lane Change Left/No/Right
- Lane Change Left/Right+No
- Lane Change Right/Left+No

For each of these models Floating Car Data and lane change labels are used to train the model on lane change classification. An illustration of how the different inputs are used, is depicted in figures 3 and 4.

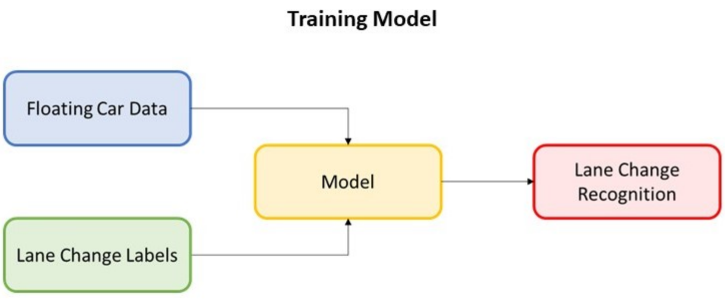


Figure 3: Schematic representation of model training

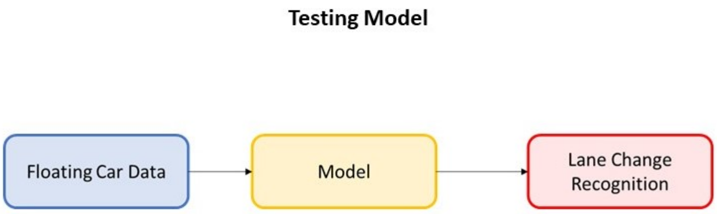


Figure 4: Schematic representation of model testing

For each of the types of lane changes, the features speed, heading, delta heading, heading difference to the previous data point, timestamp, Y-distance, and X-distance, are investigated as lane change indicators.

Lane changes were found to be recognised accurately between 51.98 % and 53.10 % through the rule-based method compared to the trajectory reconstruction algorithm. However, this method found only a total of 7 to 8% of lane changes compared to the trajectory reconstruction algorithm, and 14.3 to 26.9% compared to literature.

From the four models trained through a Random Forest algorithm, lane changes were found to be accurately recognised between 48.84% and 64.50% depending on the lane change categorisation. The most important indicators were found to be the heading of the vehicle, and the X-distance between the vehicle and the centerline of the road, in which the X-distance weighs more heavily in lane changes to the left, and heading more in lane changes to the right.

The most suitable lane change recognition model is dependent on the exact goal of the model. For cases in which the precision of the recognition is most important, thereby putting more importance on a found lane change to have actually been made as said, a model with a high precision score is required. In cases where as many lane changes as possible need to be recognised, in which case it is not critical if a few lane changes are wrongly labelled, a model with a high recall score is more suitable. In case both are equally important, a model with high f1-score and accuracy should be used. Furthermore, the most suitable model is dependent on the type of lane change aimed to be recognised i.e. whether the direction is required.

A number of limitations apply to this study which should be taken into account when considering the results. Firstly, this study is the first study in which the trajectory reconstruction algorithm by Arman and Tampere (2021) is applied outside of the original study, as well as in a different study area and country. The loop detector- and road placement for this is also executed manually, which is sensitive to errors and thereby potentially influenced the labelling.
Secondly, the hyperparameters for each model were tuned using a random search, which is a method that considers only a limited number of values within a determined range for each hyperparameter. A more thorough method would be a grid search, which considers every value. This method however is very time- and computationally demanding and for that reason not applied in this study.

For future research, the implementation of a grid search is therefore recommended in order to determine whether this leads to increased accuracy of the models. Furthermore, it is recommended to apply this research structure to a different road (section), as well as to several road sections combined, in order to research the generalisation of the models. Lastly, adding a third data source such as a camera to this research would increase the reliability of the lane change labelling, leading to more reliable results, and is therefore recommended.

Combining the findings of this study with the results from Van Ballegooijen (2019), it seems that with the current level of GPS inaccuracy, it is, at least with these methods, not possible to reach a lane change recognition rate of more than 50-64% from Floating Car Data alone. By improvement of the GPS receivers in mobile devices and navigation systems, a higher accuracy can probably be reached in the future.

1 Introduction

Throughout the last decades, traffic congestion has become an increasingly severe problem in our modern society, with especially in rush hour many drivers spending vast amounts of time stuck in traffic. The amount of congestion in The Netherlands has been increasing over the past years, with an increase of congestion intensity of 20% in 2018, and 17% in 2019 (ANWB, 2020). Not only does this congestion lead to a loss of time and money, it also comes with a significant increase in fuel consumption and levels of emissions (Treiber et al., 2008).

Multiple ways of tackling congestion are known, varying from the promotion of alternative modes of transport, to the development of automated vehicles, and policy implementations such as taxes and congestion- charge or pricing strategies. Although in the future connected and automated vehicles are expected to aid in reducing congestion through their capabilities of cooperating at a higher level, it can still take decades for the penetration rate of such vehicles to be high enough to aid traffic flow (Makridis et al., 2018). Therefore it is interesting to look at solutions implementable in the short term in order to increase traffic flow and make better use of the available road capacity.

Currently driver behaviour is a major cause of congestion, both through direct, and indirect effects, such as lane flow distribution, and lane changing behaviour. Drivers naturally follow a specific lane flow distribution, in which the capacity flow is not reached on all lanes simultaneously, causing the road capacity to be impacted (Wu, 2006). Additionally, road users often perform sub-optimal lane changes due to flawed perceptions, which can then trigger congestion (Roncoli et al., 2017). Therefore, an optimal, more equal distribution of traffic over the lanes of a highway can aid in increasing traffic flow. Especially upstream of bottlenecks, weaving sections, and on- and off-ramps, a beneficial distribution of traffic over the lanes is desired (Knoop et al., 2010).

A way in which to affect traffic distribution would be to offer personalised advises to individual vehicles, for example telling them when to switch lanes. Considering the extensive use of navigation apps and in-vehicle navigation systems, in-car advises given by those systems can offer many traffic management opportunities. This manner of advice-implementation is specifically effective as the in-car- and navigation systems are easily adaptable, and already in use. Consequently, no infrastructure changes are required for offering (personalised) advises, and this can therefor be considered a relatively cheap and easy implementable way of increasing traffic flow.

One of the biggest current challenges withholding the implementation of such personalised, in-car advice, is the fact that it is at present not possible to determine, from GPS location, with certainty in which lane a vehicle is located. Existing GPS signals, especially those used in mobile devices, have an uncertainty margin of several meters. On a road with lanes of approximately 3.50 meters width, this offset can therefore lead to a false perception of which lane the vehicle might be in. Furthermore, a disturbance of the signal can thereby also lead to the false interpretation of a lane change, when in reality the lateral change of location could be caused by GPS error or disturbance. Because of this, it is of interest to identify a way in which to recognise lane changes from Floating-Car Data (FCD); a data source of GPS traces of vehicles equipped with a navigation device.

This research aims to do exactly this, namely exploring whether it is possible to recognise lane changes from Floating Car Data.

1.1. Scope

This research focuses specifically on the recognition of lane changes from Floating Car Data on highways in The Netherlands. The Floating Car Data considered is that collected by Flitsmeister within the research area. No differentiation is made between types of road users, as all Flitsmeister users within the research area are considered. Furthermore, no differentiation is made between discretionary- and mandatory lane changes.

1.2. Research Question

The main research question of this research is defined as follows:

To what extent can lane changes be recognised from Floating Car Data?

In order to answer this question, the following subquestions need to be looked into:

- What (level of) information can be obtained from Floating Car Data?
- Which trajectory characteristics are significant in lane change recognition?

1.3. Research Approach

To be able to answer all research questions stated above, first an extensive literature study is executed to have a good understanding of the current knowledge framework, as can be found in chapter 2. After this the availability of data for this research is being looked into, which is followed by a data analysis and filtering in chapter 4. Subsequently, the available data is used for trajectory reconstruction by the matching of Floating Car Data with Loop Detector data. These reconstructed trajectories offer information on the lane a vehicle is in at each loop detector location, through which lane changes can be found for each trajectory.

This data, consisting of trajectory information and labels of whether a lane change is made, is then used in order to recognise lane changes without the use of the loop detector data. This is done by applying a rule-based lane change recognition in chapter 5, as well as by creating a machine learning lane change recognition model in chapter 6. Finally, in chapter 7 the findings are discussed and a conclusion is reached. A schematic overview of this research structure can be found in figure 1.1 below, in which the arrows indicate the order in which information/findings from previous steps are used in the next steps.

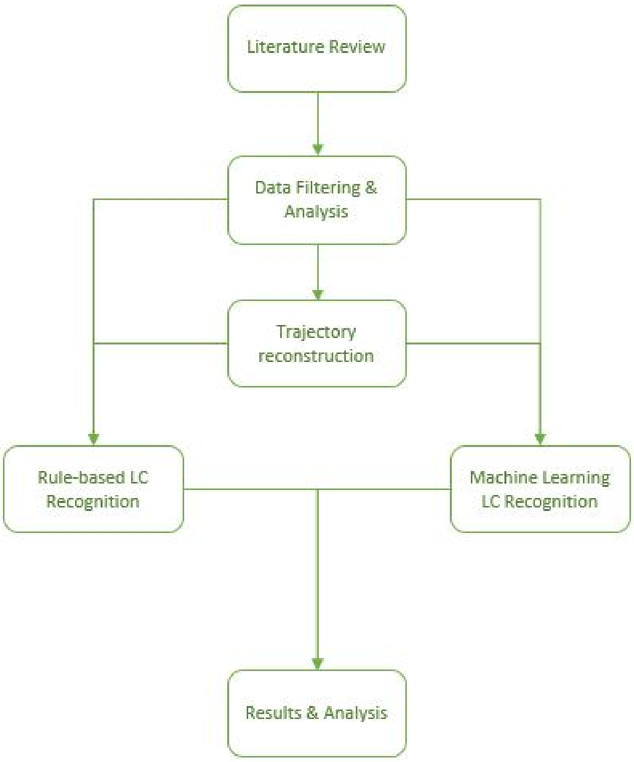

Literature Review
Data Filtering & Analysis
Trajectory reconstruction
Rule-based LC Recognition
Machine Learning LC Recognition
Results & Analysis

Figure 1.1: Research structure

The loop detector data in this research is used only as a means in order to find whether or not a vehicle makes a lane change in a road section. The goal is to then be able to recognise lane changes without the use of loop detector data. By not using loop detector data, but recognising lane changes from only the Floating Car Data, it is possible to apply this recognition of lane changes on any road, regardless of the infrastructure. Especially with the increased use of in-car navigation and navigation apps, this offers many traffic management opportunities.

2
Literature review

To research the current knowledge on, and related to, lane change recognition from Floating Car Data, a literature research is executed, looking into both lane change characteristics from a microscopic perspective, as well as GPS accuracy and the interrelated effect of these. Furthermore, past and current research on trajectory tracking related to lane changes is investigated. Lastly, researches on lane change recognition through machine learning are looked into. The combination of these different researches aids in determining the research gap and the elements important to consider in this research.

2.1. Lane change definition and characteristics

In order to be able to recognise lane changes, and understand which characteristics indicate a lane change, it is important to have a clear definition of what a lane change is. In literature previous studies on lane changes and lane change characteristics have often used diverse definitions of the beginning- and end- point of a lane change, ranging from the moment a vehicle's wheel crosses the lane boundary, to the moment the center of the vehicle has reached the destination lane. Some studies even consider the moment the driver decides to make a lane change as the moment the lane change starts (Cao et al., 2013). Due to these strongly varying definitions, characteristics of the lane change such as time required to complete a lane change vary largely among studies.

Several studies have looked into the average duration of a lane change, and thereby found differing results. Although the average comes down to 5-6 seconds for a lane change, results were found ranging from 1 to 16 seconds (Cao et al., 2013). Many factors influence this duration, such as, but not limited to, traffic conditions, the relation of the vehicle to its surrounding vehicles, and the riskiness of the move (Toledo and Zohar, 2007). Additionally, the type of vehicle also influences the lane change duration, with a lane change made by a heavy vehicle being on average shorter than that of a passenger car (Toledo and Zohar, 2007).

Lane changes occur in many different locations in the road network, but the intensity of such manoeuvres is a lot higher near network nodes and weaving sections - the places where traffic flows merge and diverge from the main stream (Arman and Tampere, 2021). On average a vehicle has been found to make 0.4 to 0.5 lane changes per kilometre, in which the number increases with density of both the origin and destination lane (Knoop et al., 2012).

One way of defining a lane change, as adopted by Li et al. (2017), is by looking at the steering angle, from which a lane change is defined between the moment of maximimum positive (negative) steering angle and the moment at which the steering angle value falls below (rises above) 10% of this maximum positive (negative) steering angle. This study looked into retrieving discretionary lane change characteristics from trajectories. In order to do so the bicycle model of steering manoeuvres was used in order to estimate driving manoeuvres from the available vehicle trajectories. Since the steering angle during discretionary lane changes is usually quite small, the bicycle model is accurate enough in representing the steering pattern.

2.2. GPS inaccuracy

Nowadays many devices, including passenger cars, trucks and smartphones, have a GPS receiver which can track the vehicle or device's location (Skog and Handel, 2009). This offers many research opportunities and the potential to track vehicle's driving movements by recording their trajectories. The GPS technology however has some drawbacks, one of the foremost being data drift and error. The accuracy of the GPS signal can be affected by multiple causes, both within the vehicle such as the receiver quality, internal filtering, or placement within the vehicle, but also by external factors such as interference, atmospheric conditions etc (Arman and Tampere, 2021). Additionally, the location accuracy can differ widely between trajectories (Arman and Tampere, 2021). Furthermore, signal loss can also be caused due to infrastructural influences such as overpasses and tunnels, since in order to get an accurate (three dimensional) positioning, the GPS needs to have a clear line of sight to at least four or more satellites. Although some receivers can also detect reflected signals, the accuracy of the GPS location can be strongly reduced in such cases (Goodall et al., 2006). The combination of these factors and reasons for errors, causes the GPS location to lack full reliability and lead to a potential offset of up to several meters.

In order to evade this unreliability, many commercial in-car navigation systems use map matching to compare trajectories and location from GPS receivers with roads as indicated on the digital maps. The most likely position on the road is then estimated. This estimate will however be a lot less accurate in cases when buildings block satellite signals in urban areas, or too little satellites are available in order to offer proper GPS location (Skog and Handel, 2009). For navigation purposes, technologies have been developed which use other sources of information such as accelerometers, gyroscopes, or odometers to increase the accuracy of the positioning (Skog and Handel, 2009).

2.3. Lane change recognition through Floating Car Data

Multiple studies have been executed on lane change recognition by use of camera, steering wheel angle, or probe vehicle data. However, to the best of our knowledge, to date, no scientific research has been done on lane change recognition from only Floating Car Data.
An investigation executed by Van Ballegooijen (2019) however has looked into lane change recognition from Floating Car Data, namely by looking at the moment at which vehicles make a lane change in a weaving section, and whether the moment at which this is done changes when offering in-car advices. For this investigation, a lane change definition is required, for which the delta heading (Δh) of a vehicle is considered, which is defined as the difference in heading of the vehicle compared to the infrastructure. The definition of a lane change is then formulated as follows, in which t_0, t_1 etc.. represent the time steps, per second, of a Floating Car Data point, and $\Delta h_{t_0}, \Delta h_{t_1}, etc$ represent the delta heading at each time step.

$$(\Delta h)\ at\ t_0,\ t_1,\ and\ t_2 < 0\ or\ (\Delta h)\ at\ t_0,\ t_1,\ and\ t_2 > 0$$

$$and$$

$$abs(\Delta h_{t_0} + \Delta h_{t_1} + \Delta h_{t_2} + \Delta h_{t_3}) >= 6$$

This definition was determined by looking at vehicle traces from Floating Car Data of vehicles which were first detected on the main road and later on the off-ramp or other road, or which entered the main highway from the on-ramp, ensuring these vehicles made at least one lane change by switching roads. Of all vehicle traces which were found to have made at least one lane change, 67% where defined as having made a lane change according to the used definition. This indicates that the definition does not cover every lane change, but it does offer a good picture of lane change characteristics (Van Ballegooijen, 2019).

When testing the same lane change definition on two road segments, the percentage of accurately recognised lane changes was found to be between 50-60% for one segment, and around 70% for the other one, indicating a majority, but not all lane changes are covered by this definition.

This study by Van Ballegooijen, 2019 is thereby a unique case in using Floating Car Data (in this case Flitsmeister data), for lane change recognition.

2.4. Machine Learning in Lane change recognition

Machine learning is of growing importance and increasingly used for all kinds of applications in many different fields, including lane change recognition, thereby offering good opportunities for analysing data further where standard methods reach their limits.

As Louppe (2014) mentions in his work 'Understanding Random Forests: From Theory to Practice', machine learning can be defined as "the study of systems that can learn from data without being explicitly programmed" (Louppe, 2014, p.2). There are several types of machine learning models, of which a clear split is made between supervised and unsupervised methods.
Several studies have been executed to recognise lane changes through machine learning. Many of these however focus on autonomous vehicles, or look into lane changing of vehicles relative to the positions of surrounding vehicles. Monot et al. (2018) for example looked into recognising the lane changing of surrounding vehicles, thereby using the speed of the vehicles and the lateral- and longitudinal positions over the previous 25 seconds. The research compares rule-based lane change recognition (using Kalman filter and probabilities) with machine learning lane change recognition (two Neural networks). The rule-based method looks at how close a vehicle is to the lane markings, in combination with its transverse speed to calculate the probability of lane changing. It was found that the neural networks were more accurate in lane change recognition than the rule-based method.

A study by Das et al. (2020) compared the accuracy of four different models namely Random Forest, Support Vector Machine, Artificial Neural Network, and eXtrem Gradient Boosting, using different vehicle kinematics (speed, longitudinal acceleration/deceleration, lateral acceleration/deceleration, and yaw rate (the angular velocity of the vehicle)), machine vision (lane position offset), roadway characteristics, and driver demographics as features. The dataset consisted of 1200 lane changes, and 2400 no lane changes, and a differentiation was also made between weather conditions. It was found that the highest detection accuracy was 95.9% when including all features in the eXtrem Gradient Booster model. When however only using vehicle kinematics, the Random Forest model was found to have the best performance. The authors thereby also advice using a Random Forest model for lane change prediction when only vehicle kinematics are available.

Furthermore, in some studies simulation environments are used to obtain data about lane changing behaviour. Dogan et al. (2011) used machine learning techniques to predict the time of an expected lane change for both straight and curved roads, looking, among others, at lane offset, lateral acceleration, and steering angle through a simulation study with 10 participants. Three machine learning methods were looked into, namely feed forward neural network, recurrent neural network, and support vector machines. The best combination of features to be used was also explored. It has been found that a lane change can be recognised up to 1.5 seconds before the centre of the vehicle crosses the lane marking. The best results found using the support vector machines. The most important inputs found are lane offset, steering angle, and time to contact with the vehicle in front.

Wang et al. (2019) used a combination of random forest with gini coefficient, and a long short term memory model to analyse and select the most important features which influence lane change behaviour, and recognising lane changing respectively. The study used data from buses on Chinese roads, and used video footage for classification. Seven features were found to be the most important, including the standard deviation of longitudinal acceleration and speed.

Schlechtriemen et al. (2015) executed a research looking into the probability of vehicles either lane following, making a lane change to the left, or making a lane change to the right. The probability was decided by use of a Random Decision Forest. Considering the fact that in traffic a large majority of all driving consists of lane-following, this research makes use of the Mixture of Experts approach in order to split the data set. The features used are the lateral state and velocity of the vehicle, and the relations of the vehicle to the surrounding vehicles, such as the distance between the vehicle and its surrounding vehicles.

2.5. Research Gap

Although numerous studies have been executed on lane changing behaviour and lane change recognition, most of these either use the interaction between vehicles as an indicator, or use data in the form of radar or video cameras. Often probe vehicles are used to learn lane change characteristics, but research is lacking on the large scale, accurate lane change recognition through GPS and/or Floating Car Data. This type of lane change recognition is of much interest since it offers the potential for real-time lane change recognition on any road, without sensors or cameras required.

Considering successful results found in multiple studies on lane change recognition through machine learning, this is considered a promising method to consider for this case.

3 Research Area

The research area considered for this research is the A27 between Utrecht Noord and knp Eemnes, as indicated in figure 3.1. This area was decided on following the availability of individual vehicle passage information from the loop detectors on this road. On a majority of Dutch highways, the loop detector data is available only in aggregated format, rather than per passage as was required for this research. This road section however offered this level of data, and following the fact that Floating Car Data could also be acquired for this road section, the area was chosen.

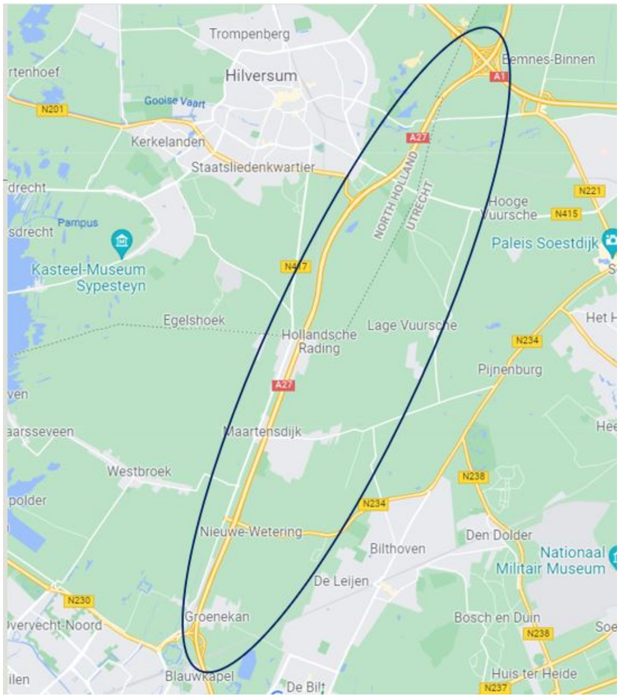


Figure 3.1: Research Area

The road of concern is approximately 13.7 km long, including several on- and off- ramps, weaving sections, and a tank station. The road has differing numbers of lanes in different sections, varying from 3 to 5 lanes. Considering the large amount of data, both the infrastructure and the traffic is only considered in Northern direction in this research.

The road was divided into homogeneous segments, dependent on the number of lanes. This led to 12 segments, as indicated in table 3.1 below. As can be seen the lengths of the segments is quite differing, as are the number of loop detector locations within each segment. The numbering of the road segments is according to driving direction, so in this case from South to North.

Road Segment	Length (in meters)	Number of lanes	Number of Loop detector locations
1	448	3	2
2	223	5	0
3	588	4	1
4	634	5	2
5	1086	3	3
6	323	4	0
7	6038	3	15
8	197	4	0
9	798	3	3
10	337	4	0
11	2466	3	5
12	495	4	2

Table 3.1: Overview of road segments

4 Data Preparation and Analysis

Large amounts of data are crucial for many researches, especially those using machine learning techniques. A common problem with large data sets however is the lack of quality, or missing data. It is therefore critical to analyse the available data prior to executing the research in order to be aware of the condition of the available data. For this research both Floating Car Data and loop detector data were made available. These data sets were collected for overlapping days and location. In the following sections the data of both these sources will be described and the quality analysed. Furthermore, a trajectory matching algorithm will be applied with the data in order to create a data set labelled according to lane changes.

4.1. Floating Car Data

With the increased use of in-car navigation systems and mobile navigation apps such as Flitsmeister, comes the advantage that large amounts of trajectory data is being collected. This data, collected from individual vehicles is known as Floating Car Data (FCD), and can offer insights into the driving patterns of drivers. The FCD offers GPS locations of the user with a 1 HZ frequency, as well as speed of the vehicle for each timestep, and an (anonymous) unique session ID.

In the Netherlands Flitsmeister is one of the most known and used companies collecting Floating Car Data, with 1.8 million people making use of their services (“Floating car data via flitsmeister”, n.d.). This in turn means that the trajectory data of a substantial percentage of road traffic is gathered by them. Flitsmeister offers the anonymous Floating Car Data to both companies and individuals in order for them to utilise the data for all types of analysis. For this research such Flitsmeister data is used as the source of Floating Car Data.

The Flitsmeister Floating Car Data for this research was collected of the highway A27 between Utrecht Noord and Knooppunt Eemnes for a period of 17 days from 21/06/2021 to 07/07/2021. Per day, trajectories of around 45000 vehicles were collected. The region for which data was collected can be seen in figure 4.1.

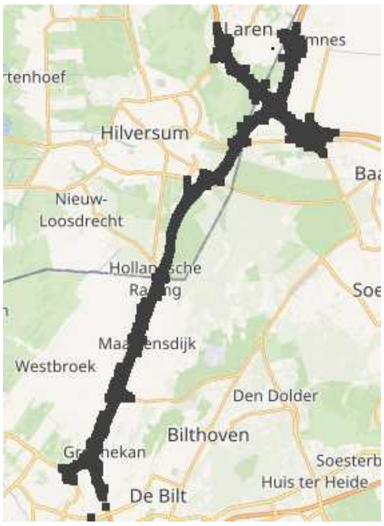


Figure 4.1: Area of Flitsmeister data

As mentioned before, GPS data comes with the problem of inaccuracy of up to several meters. Therefore, trajectories directly originating from the Floating Car Data can not be considered completely reliable, and it cannot be said with certainty whether a vehicle is actually located at the precise location the GPS indicates. Due to this uncertainty, lateral movements indicated by the GPS can not be simply relied on as indicating a lane change.

In order to get a good feeling of the quality of the Floating Car Data, several analysis have been done both before and after the filtering of the data.

4.1.1. Data Analysis

In order to be aware of the quality of the Floating Car Data offered by Flitsmeister, an initial analysis has been executed. For this analysis the raw data has been used, excluding only data that falls outside of the study area.

Considering the fact that GPS data is rather inaccurate and has quite a large error margin, an initial step in the analysis was the visualisation of some randomly chosen vehicle trajectories. Figure 4.2 depicts the trajectories of a number of vehicles driving in road segment 7. The separation of the three lanes in the road segment are illustrated by the stippled lines, and the trajectories are plotted according to the distance of each data point to the centerline of the road segment. On the Y-axis the longitudinal location is indicated in meters (Rijksdriehoeks coordinates, the spatial reference system in The Netherlands). In this plot vehicles of which the trajectory has an offset of more than 30 meters from the centerline are excluded.

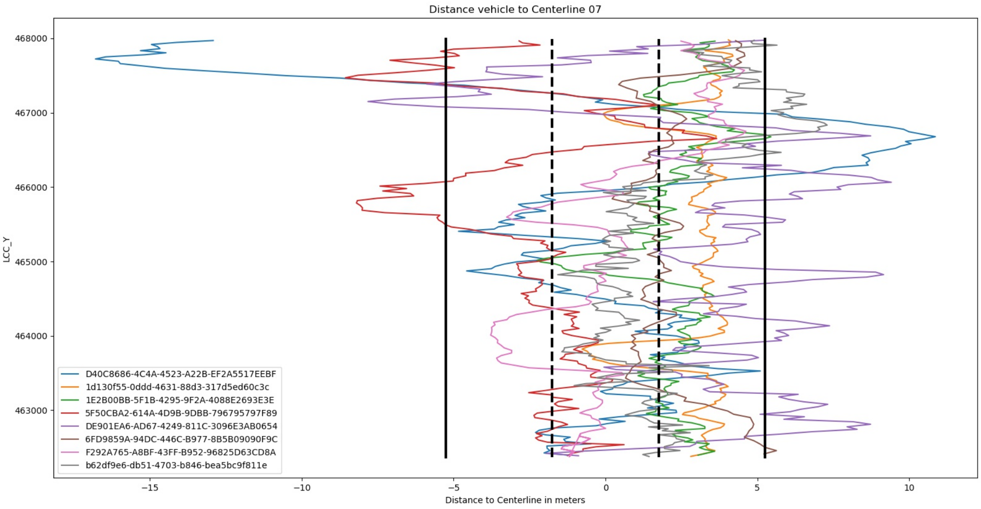


Figure 4.2: Plot of vehicle trajectories

From the figure it can be seen that quite some parts of several of the trajectories are located outside of the road, in many cases probably caused by the GPS error, especially since the vehicle several times leaves and ’re-enters’ the road segment. A trajectory that stands out in this figure is the light blue one (Session ID D40C8686-4C4A-4523-A22B-EF2A5517EEBF). A large offset from the road can be seen near the end of the road segment. An initial thought for this error could be that the vehicle left the highway and then passed perpendicularly over or underneath the road. However, considering the fact that the road sections are homogeneous, and this segment therefore does not contain an off-ramp, this does not seem to be a realistic option. It should however be taken into account that there might also be other road users, for example on parallel roads, who use Flitsmeister, whose signals can be received as if on the highway and thereby leading to noisy data. Many of these cases can be filtered out when pre-processing the data, for example by excluding certain speed ranges, especially when deviating strongly from the other traffic and so from the traffic state at that time. This situation does however clearly indicate the difficulty associated with the GPS unreliability, making it difficult to know whether such a trajectory spike is caused by GPS error or other causes.

A second analysis of the data is done by looking at the percentage of data that falls within each lane in a segment. As can be seen in figure 4.3, most data points fall within the road. Segment 9 can be seen to have a higher percentage of data points falling outside of the road, which can most likely be explained by the fact that the segment is curved, which is known to cause GPS distortion. A table with all the exact percentages can be found in appendix A.2.

Another notable fact is that a vast majority of the data points are located on the three left lanes, even in road segments consisting of 4 or 5 lanes. This can be explained by the fact that the right lane or two right lanes are in all segments off-ramps or weaving sections, which therefore do not contain through traffic.

It must be noted that due to the length of segment 7 and thereby its corresponding processing time this segment was excluded from this analysis.

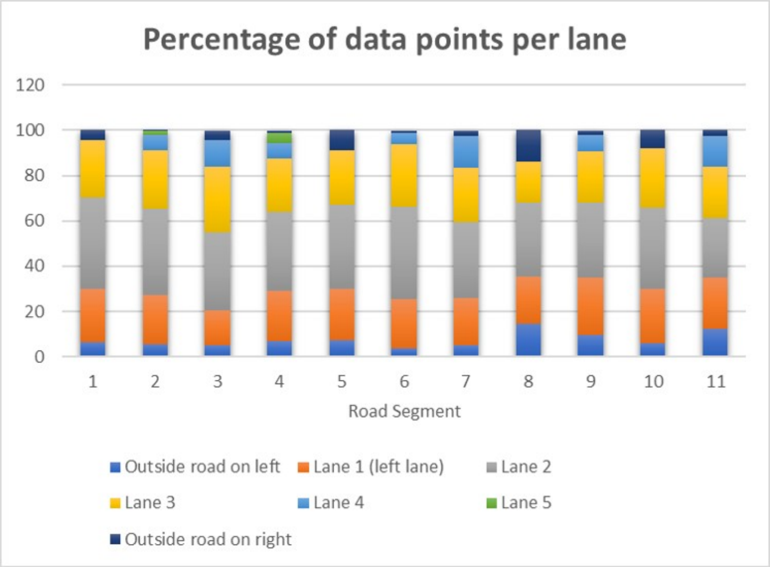


Figure 4.3: Percentage of data points per lane

4.1.2. Filtering of data

After collecting the data from its users, Flitsmeister provides all data from within the boundaries of the required geofence. The geofence, which additionally to a part of the A27 used in this research, also contains a small part of the A1, can be found in section 4.1.

The data within the geofence is unfiltered, and the data set consists of all data produced by the users within the specified area. The remaining trajectories within this region therefore still contain trajectories which are not desired in this research such as outliers and incomplete trajectories. To ensure high quality data, the data set is filtered according to the following definitions:

- Only trajectories within the study area (the A27 between Utrecht Noord and knp Eemnes) are considered, so parts of trajectories outside the X- and Y- coordinates of the area of interest are excluded. Since the geofence lies rather broadly around the A27, and even includes parts of the A1, the trajectory pieces lying outside of the study area are excluded by this manner of X- and Y-coordinate limiting. This means that any trajectories starting in, but ending outside of the study area are simply cut off at the edge of the section of interest, ensuring no information from within the study area is lost. The study area in this case can for some analysis be solely a road segment.
- Trajectories containing a proportionally large number of data points. A large number of data points was in this case defined as anything more than the number of data points corresponding with a vehicle driving 70km/h on the specified segment. By following this selection, the trajectories of both congestion, and vehicles which are present in the study area for an exceptional amount of time, e.g. those stopping at a fuel station, are filtered out.
- Trajectories containing a very small number of data points. The threshold for this amount was set as the number of data points that a vehicle driving 150km/h would have in the determined segment. This filters out both incomplete trajectories, as well as vehicles driving exceptionally fast.
- Furthermore, the Floating Car Data set is filtered to include only trajectories which are able to be reconstructed according to a method created by Arman and Tampere (2021). This process and requirements for such reconstruction will be explained in detail in section 4.4.

Additionally, to limit the scope of this research, only traffic driving in Northward direction is considered. The trajectories are filtered on monotonically increasing Y-coordinates, which due to the straightness and direction of the road placement, and the short road segments used, is considered an acceptable method. For larger road segments, or roads placed differently directionally, this filtering method could possibly remove crucial information and would not be recommended.

4.2. Loop detector data

On many Dutch highways loop detectors are placed every few hundred meters in order to measure and register the passing traffic. These inductive-loop traffic detectors make use of an electrical circuit and measure the change in the magnetic field when a vehicle passes over it. For every passage on every lane, the speed, vehicle size, and time of passage are recorded, offering a good overview of the traffic state on the highway. The loops record every individual vehicle passage, after which it is often aggregated to get the average speed/traffic state of the road.

For this research dissagregated loop detector data is collected from the same location as the Flitsmeister data is collected, namely the A27 between Utrecht Noord and Knooppunt Eemnes. This data was obtained for a period of 19/06/2021/ to 06/07/2021, and offers information on individual vehicle passages over the loops. In total this study area contains 33 loop detector locations, an overview of which is given in table 4.1 below. As can be seen, the detectors are not equally divided along the segments, and several segments do not contain any loop detectors.

Segment	Number of loop detector locations
1	2
2	0
3	1
4	2
5	3
6	0
7	15
8	0
9	3
10	0
11	5
12	2

Table 4.1: Number of loop detector locations per segment

4.2.1. Data analysis

In the graph below (figure 4.4) the number of passages per loop detector are indicated for one day (27/06/2021). This offers an idea of the distribution of traffic over the lanes. An overview of all passage counts per loop detectors for all days can be found in the appendix A.3. The lane numbering starts with 1 for the left lane, and increases numbers towards the right-most lane.

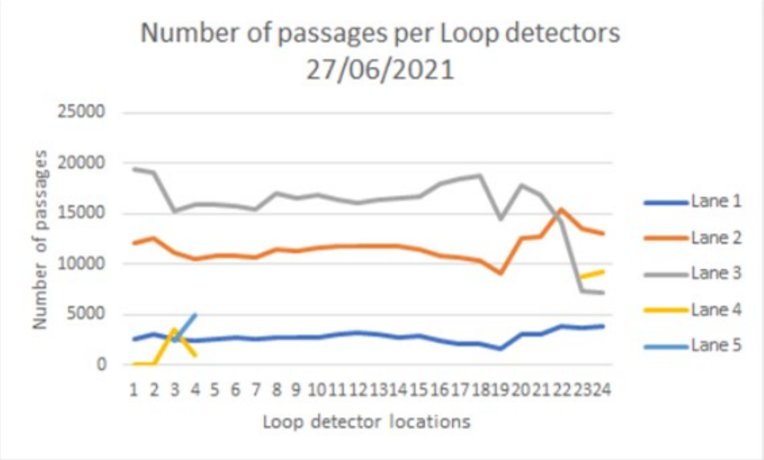


Figure 4.4: Number of passages 27/06/2021

4.3. Centerline

In order to analyse the positioning of the trajectory data points in respect to the road, the location of the road and the lanes have been identified. This has been done through manually acquiring the centre of the road sections through QGIS according to the most recent aerial picture of PDOK. Through this the coordinates of the centerline of each road segment were created, after which the lanes and their corresponding separators could be determined according to the known width of the Dutch highway lanes. For each road segment the number of lanes are known, and thereby the road can be positioned.

4.4. Trajectory reconstruction

As has been demonstrated, the vehicle trajectories deduced from the Floating Car Data do not accurately represent the actual trajectory travelled by the vehicle. Arman and Tampere (2021) have developed an algorithm which reconstructs the trajectories up to lane-level by data fusion of Floating Car Data and loop detector data. The passages of the vehicles over the loop detectors are matched with the trajectories from the Floating Car Data by looking at the passage time and passage speed of each loop detector as well as that of the trajectory at the loop detector position. Through this method the actual lane a vehicle is in can be deduced.

The first step in this reconstruction algorithm is the filtering of the trajectory data. Trajectories that have a temporal interruption of more than 10 seconds, or a spatial interruption of more than the length the vehicle travels in 10 seconds at its average trajectory speed, are excluded. After which the trajectories for which a map-matching algorithm can not be applied are excluded, and finally any trajectory which is not long enough to pass over at least two loop detectors is also excluded. From the left over trajectories, a map-matched version is created through an algorithm by Quddus et al.

Secondly, these map-matched trajectories are slightly corrected to get rid of minor errors, by interpolation using the heading and speed of the vehicle before and after the interruption. Furthermore, errors due to sharp deceleration or zigzag movements due to GPS error are corrected.

Next, the data fusion is being executed combining trajectory data and loop detector data. The trajectory data offers the information when the vehicle passes a loop detector location in its path, while the loop detector can then determine the driving lane of the vehicle. By using the vehicle length as control variable (this value stays the same the entire trajectory), the driving lane of the vehicle can be determined throughout its whole trajectory. This thereby also offers the information of sections in which a lane change must have taken place, as it is known when a vehicle passes a loop detector on a different lane compared to the lane of the loop detector it previously passed.

Since the passage time recorded for a vehicle over a loop can differ from -1 to +1 second, it is possible that other vehicles also pass the loop detector in this time-frame. In order to still be able to identify a vehicle and execute the matching, the speed and passage time of the vehicle is compared to the speed and passage time of all loops in a loop detector location.

In the original research by Arman and Tampere (2021), in order to check the accuracy of the smartphone trajectory data with the loop detector data, a probe vehicle using d-GPS was used. Similarly to the smartphone data, this data source offers time and speed of the moment the vehicle passes the loop detector location. However, since the data from the d-GPS device is very accurate, it is assumed to be the ground truth, and the passage time and speed can be compared to that of the smartphone trajectories. It was found that the statistical distribution of passage time difference between the d-GPS trajectories and the corresponding loop detectors, and the passage time difference between the smartphone trajectories and loop detector data are identical. This means that the passage time of every vehicle is detectable with a good approximation based on the trajectory data (Arman and Tampere, 2021). Furthermore, it is also shown that the error of the speed information for both the loop detectors, and the trajectory data are negligible, as are the error of the measurements of the length of the vehicle.

As a last step in the trajectory reconstruction algorithm, the trajectory is reconstructed in between the loop detector locations. This is done by forward and backward modifications in-between two successive loop detectors, assuming the speed and heading of the trajectory information are accurate. The final trajectory path is then the weighted average of the forward and backward reconstructed path in between two consecutive loop detectors.

This study was originally executed on the Antwerp ring in Belgium, and was validated by using videos made by drones and road CCTVs at two sections of the test network. Furthermore, a vehicle with differential-GPS (d-GPS) and nine smartphones using the Be-Mobile application was also used to create trajectory data serving as ground truth.

The results of the study showed that this method is very reliable, with a success rate of 96.86% for the data fusion part in which the trajectory data is matched with the loop detectors. The fully reconstructed trajectories were found to be successfully located on the correct driving lane 93.22% of the time when validating according to smartphones in the probe vehicle, and 90.68% of the time when validating with the drone's video recordings. This leads to an overall accuracy of over 90% in determining the driving lane of vehicles (Arman and Tampere, 2021).

4.4.1. Application in this study

Considering the high success rate of this algorithm, and the availability of similar data in our research area, it is possible to apply the same trajectory reconstruction to our study. By doing so, the lane in which a vehicle drives becomes clear, and it is known whether or not a vehicle made a lane change in a road section between two consecutive loop detector locations, and if yes from which origin to which destination lane. Such information can be considered a ground truth, and a labelled data set can be created, which can then be used in order to deduct lane change characteristics from the trajectories. From now on, the ground truth considered in this study relates to the information of which lane the vehicle is found to drive in according to the data-fusion of the Floating Car Data and the loop detector data. For each road section between two consecutive loop detector locations it is then deduced whether or not a lane change has been made. In order to do so, QGIS is used for manually indicating the location of the road in the study area, as well as the loop detector locations within each section.
Figure 4.5 shows a schematic depiction of the use of the two data sources for the creation of the lane change labels.

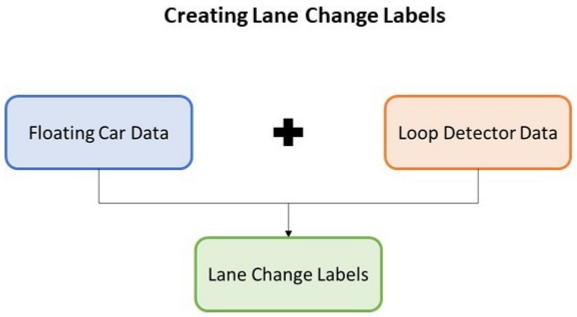


Figure 4.5: Schematic representation of data usage step 1

This study is the first study in which the algorithm by Arman and Tampere (2021) is applied in a different setting. Although the setting of that study and this current study are quite similar, there are some differences, which could potentially affect the success of the algorithm in this study. One of such differences is the fact that the loop detector locations in the original study are placed on average every 400 meters, while in this current study they are placed at a much more varying distance from each other. Furthermore, this study area is located in The Netherlands, while the original study was set in Belgium. Although the differences should not be significant, there might be variations, for example in the quality of the loop detector data, or the amount of objects disturbing the GPS signal around the highway. It is also possible that driving style differs slightly between the countries which can impact the type, location, and number of lane changes executed.

On top of this, it must be considered that the location of the road and the loop detectors are manually placed, which is a delicate process in which a wrong placement can have quite an impact on the results. Especially in high traffic densities, many vehicles pass each loop detector, and even a slight misplacement could potentially lead to wrong matching. The loop detector placement is therefore executed very carefully, according to information on their location received from Rijkswaterstaat together with the loop detector data.

During the matching process not all loop detectors available in the road section were used for the matching with the trajectories. This was caused by human error due to which a few detectors were missed, and although in a few cases it increases the distance between consecutive loops, as well as reduces the number of labelled road sections, it is not expected to have a big negative impact on the trajectory reconstruction. Especially taking into account the relatively large study area considered, as well as the large amount of data available, the missing matching points are not expected to significantly impact the ground truth found.
In the figures below some examples of reconstructed paths in this study can be seen for a small part of the study area. The first picture shows the road configuration, and the placement of the loop detectors (the coloured circles). The next four images indicate the original trajectory according to the Floating Car Data depicted in green, and the reconstructed trajectories depicted in red. In the first two images a lane change is being made between the first two loop detector locations, while in the last two images no lane change is being made. In all figures the road edges are shown as pink lines.

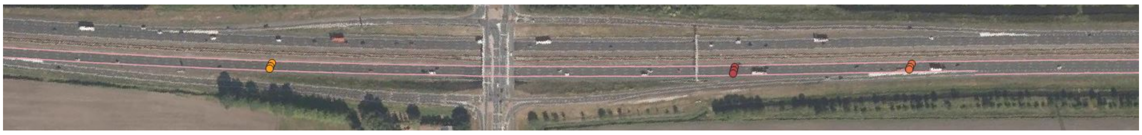


Figure 4.6: Road example including loop detectors

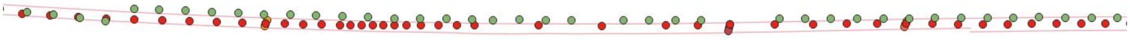


Figure 4.7: Reconstructed trajectories Yes Lane Change

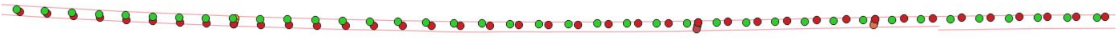


Figure 4.8: Reconstructed trajectories Yes Lane Change

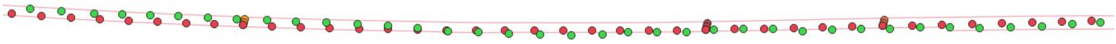


Figure 4.9: Reconstructed trajectories No Lane Change

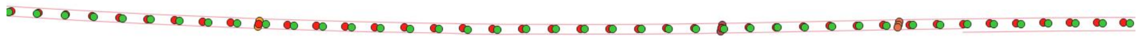


Figure 4.10: Reconstructed trajectories No Lane Change

The reconstructed trajectory in the first figure (figure 4.7) clearly shows that the original trajectory was located outside of the road. The reconstructed trajectory however is placed neatly within the road boundaries. From the reconstructed trajectory, it can also be seen that according to the coupling with the loop detectors, the vehicle was driving in the second (middle) lane at the first loop detector location, but moved to the left-most lane at the second and third loop detector locations. This indicates a lane change has been made in the road section between the first and second loop detector locations.

In figure 4.8 another example is depicted of a reconstructed trajectory in which a lane change is being made. This trajectory clearly did not need to be corrected as much as the previous one, but is still adjusted slightly. These examples, and their different levels of adjustment needed, demonstrate the difference in GPS accuracy between trajectories.
Figure 4.9 and 4.10 show similar adjustments, this time for trajectories in which no lane change is made. It can be seen that the reconstructed trajectory is matched to loop detectors on the same lane throughout the three loop detector locations, and so no lane change is being made.

In the figures below two more examples are shown of trajectories that have been reconstructed. The first figure represents four trajectories of Floating Car Data which have, by the trajectory reconstruction, been found to not make a lane change. Figure 4.11 depicts the trajectories on the road as found by the Floating Car Data. Figure 4.12 depicts the reconstructed trajectories on the road of these same vehicles. Here again it can be seen that some trajectories are corrected quite rigorously, while others are only slightly altered.

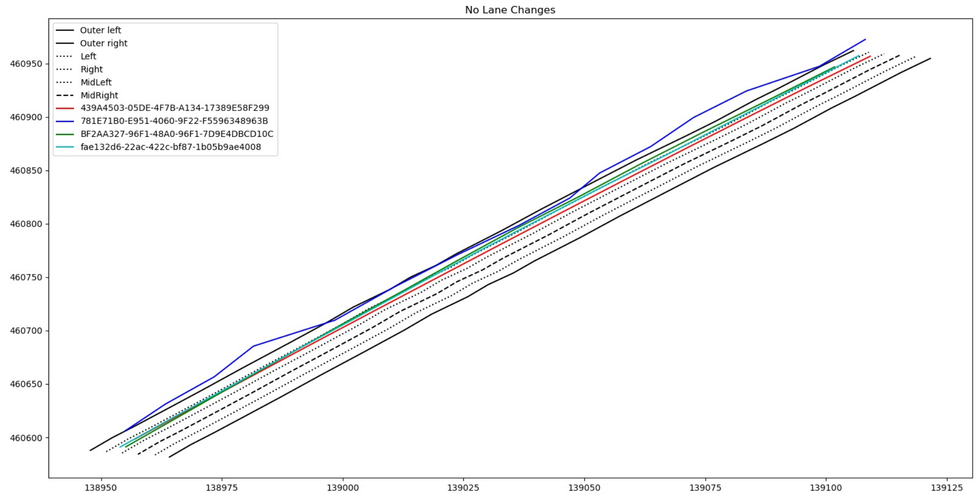


Figure 4.11: Original trajectories making no lane change

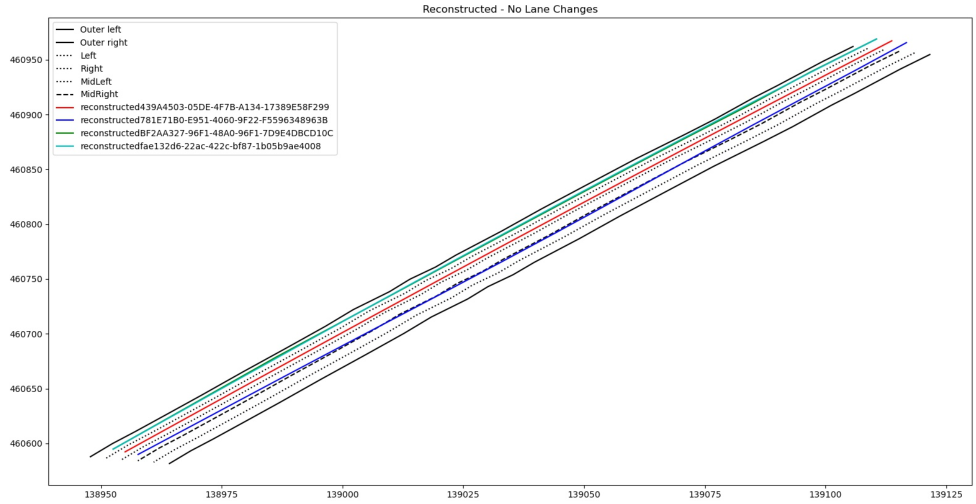


Figure 4.12: Reconstructed trajectories making no lane change

Next, the same is represented for a few trajectories which are found to have made a lane change in this road segment. Figure 4.13 depicts the original Flitsmeister trajectories, while figure 4.14 shows the reconstructed version of the same trajectories.

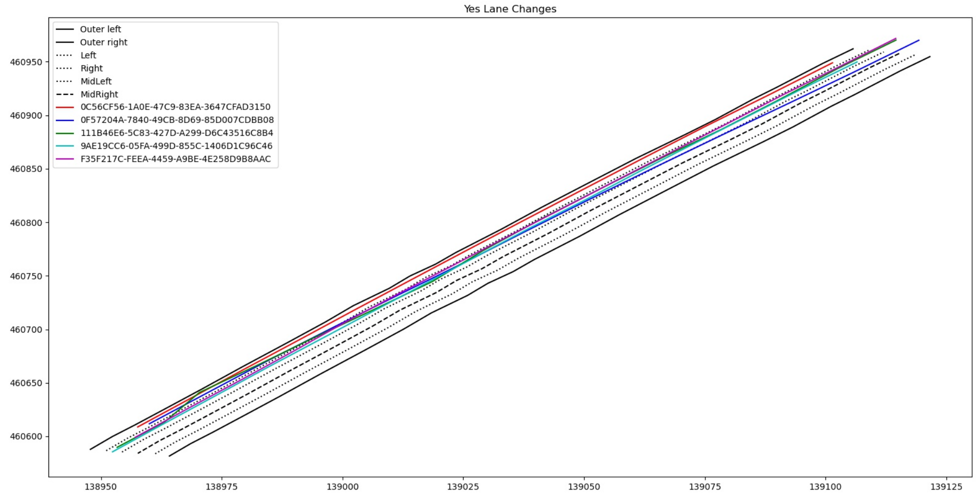


Figure 4.13: Original trajectories making a lane change

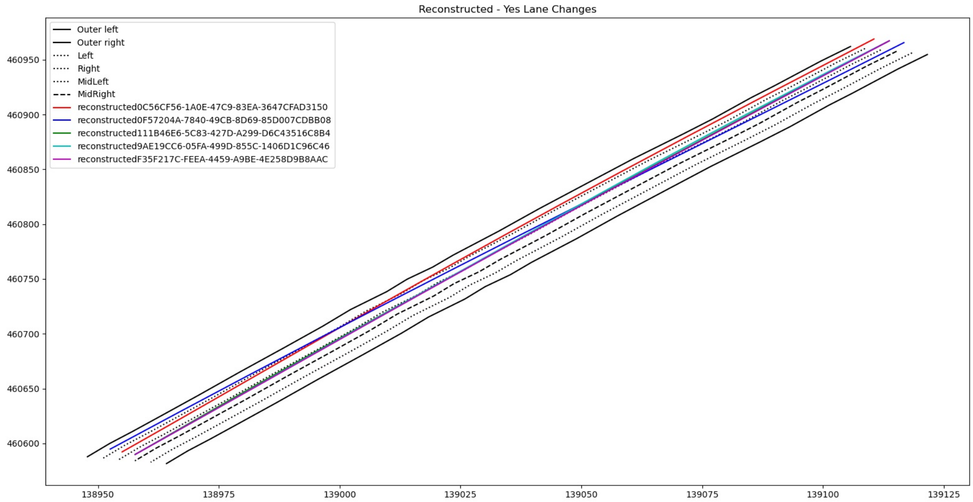


Figure 4.14: Reconstructed trajectories making a lane change

Due to the large number of trajectories, and the processing time of the algorithm, the trajectory reconstruction has only been applied to the trajectories of 26/06/2021 until 04/07/2021. Furthermore, the data of the 30th June 2021 is excluded in the reconstruction due to the fact that this data contained abnormalities compared to the other days.

4.5. Final Overview of Data

After filtering the original data, and applying the trajectory reconstruction algorithm by Arman and Tampere (2021), a reduced number of trajectories remain available for the research. The filtering criteria reduce the number of trajectories by around 60%. Although this might seem like a very large amount, it must be taken into account that only traffic driving in Northward direction is considered, meaning already around 50% of the filtered data consists of traffic driving towards the south.
An overview of the number of trajectories per day, per segment, both before and after filtering can be found in appendix A.4.

4.6. Data set creation

For each road section between two consecutive loop detector locations, a data set can be created with labels for each trajectory according to whether or not a lane change is made by this vehicle in the specific road section. It is also possible to specify whether the lane change is made towards the left or right, and how many lanes are changed.

Since each road segment consists of a large number of data, two locations are first decided upon for which to create these data sets. The first location decided on is the road section between the two loop detector locations in segment 4, which has a distance between them of approximately 400 meters. The road section consists of 5 lanes, of which 2 are exit lanes in the second half of the section. Furthermore the most right-hand lane comes from a gas station. This ensures that lane changes will be made by vehicles that either enter the section from the gas station, or exit the road through the exit lanes. Lastly, the section is rather straight, which would reduce extreme GPS error due to road curvature. These circumstances assure this section is a suitable road section on which to execute this study.

The second road segment used for analysis is the road section between the first two loop detector locations of segment 5. This road segment has 3 lanes, all of which are through-lanes. The road section is approximately 500 meters long, and also quite straight. By choosing this second road section which does not contain entry- or exit lanes, a representative combination of types of road sections and accordingly, types of lane changes made is reached in combination with the first chosen road segment.

For both of these sections, a labelled data set is created, indicating whether or not a lane change is made for each vehicle in each road section.
In addition to the original data collected by Flitsmeister, the heading, heading difference between the previous and current data point, and heading difference to the centerline are added for each trajectory. The formulas used to calculate these headings are based on Bullock, 2007, and can be found in the appendix A.5

Furthermore, the (horizontal) distance to the centerline of the segment is added as x-distance, and the longitudinal distance from the beginning of the road section in consideration to the data point is added as y-distance. All of these additional features can be used for analysis and lane change recognition.

5
Rule-based Lane Change Recognition

In this chapter the rule-based lane change definition by Van Ballegooijen (2019) will be investigated for the data from the A27. The found lane changes will be compared to those found through the trajectory reconstruction algorithm, as well as to the number of lane changes as expected according to literature. A schematic overview of this is shown in figure 5.1.

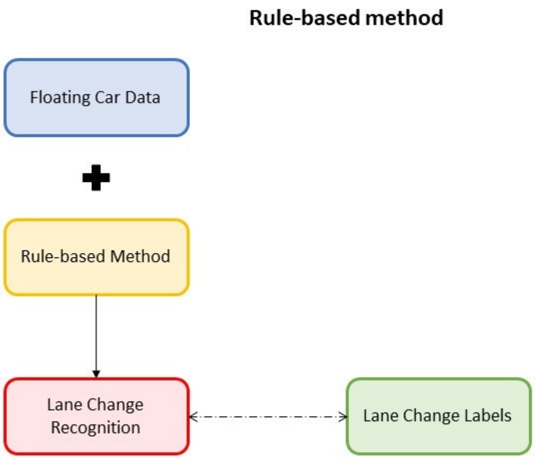


Figure 5.1: Schematic representation of rule based method

5.1. Original definition

As mentioned in section 2, a research conducted by Van Ballegooijen (2019) has found that a large number of lane changes can be identified by looking at the delta heading of vehicles over several consecutive time steps using the following definition, in which the delta heading represents the difference in heading between the vehicle and the infrastructure, measured in degrees. t_0, t_1 etc.. represent the time steps, per second, of a Floating Car Data point, and $\Delta h_{t_0}, \Delta h_{t_1}, etc$ represent the delta heading at each time step.

$$(\Delta h)\ at\ t_0,\ t_1,\ and\ t_2 < 0\ or\ (\Delta h)\ at\ t_0,\ t_1,\ and\ t_2 > 0$$

$$and$$

$$abs(\Delta h_{t_0} + \Delta h_{t_1} + \Delta h_{t_2} + \Delta h_{t_3}) >= 6$$

In the original study, this rule was found to be successful in recognising lane changes around 50-70% of the time, depending on the road section it was applied to.

In order to examine the replicability of the research, as well as the comparison to the lane change labels found through the trajectory reconstruction algorithm by Arman and Tampere (2021), it is interesting to look at how lane changes are recognised in this research according to the above mentioned definition, and whether or not the lane changes recognised correspond with the lane changes labels found according to the matching with the loop detectors.

In order to compare the two lane change definitions, the filtered data set is used, for which the labels of lane change / no lane change according to the ground truth found by matching the trajectories with the loop detector passages are known. Two road sections are chosen for which the data is looked at, namely 'location 1' - the road section between the two loop detector locations in segment 4, and 'location 2' - the road section between the first two loop detector locations of segment 5. The choice for these case study locations has been previously explained in section 4.6.

When applying this definition of a lane change to the data of our research, and comparing it with those found using the matching method with loop detectors, the following results are found for location 1:

YES lane change Loop detector method	792	15597
NO lane change Loop detector method	648	16367
	YES lane change Delta heading method	NO lane change Delta heading method

Table 5.1: Overview recognized lane changes according to different methods for location 1

From this distribution of determined lane changes, a total percentage of agreed lane changes according to both methods of 2.37 %, and a total percentage of agreed No lane changes of 48.99 % are found.

When considering the loop detector method as the ground truth, the accuracy of the rule based method can be calculated as explained in appendix A.6. Taking into account the unbalanced size of the group, a balanced accuracy of 53.10% is found.

When applying the same lane change detection rule for location 2, and comparing the results with the lane changes found by the loop detector matching method, the following results are found:

YES lane change Loop detector method	625	15495
NO lane change Loop detector method	624	18127
	YES lane change Delta heading method	NO lane change Delta heading method

Table 5.2: Overview recognized lane changes according to different methods for location 2

From which it can be found that a total percentage of agreed lane changes according to both methods of 1.79 %, and a total percentage of agreed No lane changes of 51.98 % are discovered.

Looking at the balanced accuracy score for this location, a value of 51.98% is found.

5.1.1. Analysis

From the results for both of the locations it can be seen that the rule-based lane change method by which lane changes are recognised by looking at vehicles heading difference compared to the infrastructure heading, not a lot of lane changes are found. Out of a total of 33404 trajectories considered in location 1, only 1440 are found to have made a lane change according to this rule (4.31 % of traffic). The method finding lane changes according to the matching with the loop detector data however finds 16389 lane changes (49% of traffic). For location 2 a similar distribution is found with 1249 out of 34871 (3.58%) trajectories found to have made a lane change according to the delta heading rule, and 16120 (46.2%) lane changes found making a lane change by matching with the loop detectors. This is a large difference in number of lane changes found.

Considering that on average vehicles are found to make 0.4 to 0.5 lane changes per kilometre (Knoop et al., 2012), around 0.16 to 0.2 lane changes are expected per vehicle at location 1 (400 meters), and 0.2 to 0.25 lane changes per vehicle at location 2 (500 meters). Since the road section of location 1 has a total of 33404 vehicles considered, 5344 to 6680 lane changes are expected. In the road section of location 2, 34871 vehicles are evaluated, leading to the expectation of 6974 to 8717 lane changes. An overview of this comparison can be found in table 5.3.

	Location 1	**Location 2**
Expected nr of lane changes	5344 - 6680	6974 - 8717
Rule Heading difference to road infrastructure:		
Found nr of lane changes Rule based method	1440	1249
Percentage of lane changes found	21.6 - 26.9 %	14.3 - 17.9 %
Matching method to loop detectors:		
Found nr of lane changes matching to loop detectors	16389	16120
Percentage of lane changes found	245.3 - 307.7 %	184.9 - 231.1%

Table 5.3: Number of expected and found lane changes per road section

From this analysis, it is found that the rule looking at lane changes according to the consecutive heading differences recognises around 14-27% of the expected lane changes, depending on the road section. The method matching trajectories to loop detector data on the other hand finds approximately 1.85-3 times the number of lane changes expected.

When considering the lane changes found according to the loop detector data as the ground truth, as is intended in this study, the rule-based method is found to accurately recognise between 51.98% and 53.10% of the lane changes, depending on the location.

In order to more deeply analyse the possible cause of the discrepancy in number of lane changes found, the direction of lane change is looked into for the original definition. It is found at location 1, that 57.75% of all lane changes made to either direction (so excluding No lane change) are made to the left. Out of these, only 4.1% is recognised by the rule-based method. For lane changes to the right, this value lies at 5.83%.

For location 2, similar results are found, where out of all lane changes according to the loop detector method (Yes lane change), 65.39% of the lane changes were made to the left. Out of these, only 3.4% were recognised by the rule-based method. From the lane changes to the right (34.61% of Yes lane changes) 4.79% are labelled as a lane change by rule-based method.

It therefore seems that the rule-based method does not recognise lane changes to a specific direction more accurately than to the other direction.

5.2. Adapted definition

Since in literature it has been found that the duration of a lane change can vary strongly depending on the definition, it is interesting to look into the effect of lane change recognition when considering the heading at an additional time step. This is done by adjusting the lane change definition to the following, in which both parts of the definition consider an additional time step:

$$(\Delta h)\ at\ t_0,\ t_1,\ t_2,\ and\ t_3 < 0\ or\ (\Delta h)\ at\ t_0,\ t_1,\ t_2,\ and\ t_3 > 0$$

$$and$$

$$abs(\Delta h_{t_0} + \Delta h_{t_1} + \Delta h_{t_2} + \Delta h_{t_3} + \Delta h_{t_4}) >= 6$$

Which leads to the following results for location 1, with a a total percentage of agreed lane changes according to both methods of 3.07 %, and a total percentage of 48.75 % of agreed No lane changes:

YES lane change Loop detector method	1024	15365
NO lane change Loop detector method	732	16283
	YES lane change Delta heading method	NO lane change Delta heading method

Table 5.4: Overview recognized lane changes according to different methods

For location 2 the following results were found with a total percentage of agreed lane changes according to both methods of 1.85 %, and a total percentage of 52.02 % of agreed No lane changes:

YES lane change Loop detector method	644	15476
NO lane change Loop detector method	610	18141
	YES lane change Delta heading method	NO lane change Delta heading method

Table 5.5: Overview recognized lane changes according to different methods

5.2.1. Analysis

Compared to the original definition, slightly more trajectories are labelled as making a lane change with the adapted definition for both location 1 and 2. Overall, the total percentage of lane changes also increased, although very slightly, from 51.36% to 51.82% for location 1, and from 53.77% to 53.87% for location 2.

When again comparing the number of lane changes found with the lane changes expected according to literature, similar results are found as in the original definition, namely a large difference between the two methods. The percentage of lane changes found according to the rule of heading difference compared to the road infrastructure covers around a quarter of expected lane changes at location 1, while only around 14-18% in location 2. The method of matching the trajectories with the loop detectors obviously remains unchanged and still finds two to three times the expected lane changes. An overview of the findings is shown in table 5.6.

	Location 1	**Location 2**
Expected nr of lane changes	5344 - 6680	6974 - 8717
Adapted rule heading difference to road infrastructure:		
Found nr of lane changes Rule based method	1756	1254
Percentage of lane changes found	26.3 - 32.9 %	14.4 - 18 %
Matching method to loop detectors:		
Found nr of lane changes matching to loop detectors	16389	16120
Percentage of lane changes found	245.3 - 307.7 %	184.9 - 231.1%

Table 5.6: Number of expected and found lane changes per road section

5.3. Conclusions on rule-based lane change method

The lane change definition as defined by De Verkeersonderneming (2019) was found to accurately recognise lane changes in 50-70% of the time in its own study area. When applying the same definition to our study area, a correct lane change recognition rate of around 52% was reached when comparing with the ground truth found by matching trajectories with loop detector passages. This result falls within the same range as the original study, albeit on the lower end. It must be taken into account however, that the majority of the agreed-on lane changes were lane changes not made ('no lane change'), which also consist of 96.4% of all lane changes, and therefore naturally have a higher probability of being agreed on. Furthermore, in the original study, the 50-70% of lane changes found, were lane changes found out of all lane changes made ('yes lane change'), so excluding all 'no lane change'.

Considering also that when compared to the number of lane changes expected in these section, only 14-33% of lane changes are found by this method, this method does not reach the same level of accuracy as in its original study.

Since there is room for improvement of the lane change recognition, it is of interest to look into other possible indicators of lane changes. In the next chapter this will be done through the use of machine learning.

6 Automated Lane Change recognition

6.1. Goal of Method

The ultimate goal of this research is to find out whether it is possible to recognise lane changes from Floating Car Data as only source. In this chapter that is done by the use of machine learning, through which a model is created for lane change recognition.
In the previous chapter it was found that using a rule-based method looking into delta headings at subsequent time steps has a recognition rate of around 52%. By looking into additional features deducted from Floating Car Data (such as X-distance, speed, etc..), other indicators of lane changing may be recognised, which may result in higher lane changing recognition rates. In order to determine the most important indicators, as well as to create a model for lane change recognition, a random forest classification algorithm is used. Through this algorithm a model is trained on (data deduced from) the Floating Car Data of each vehicle, and together with a label indicating lane changes per road segment, the model is trained to recognise lane changes and the presence of indicators for lane changes.
A schematic overview of the training and testing of the models, as well as the data used for this is shown in figures 6.1 and 6.2.

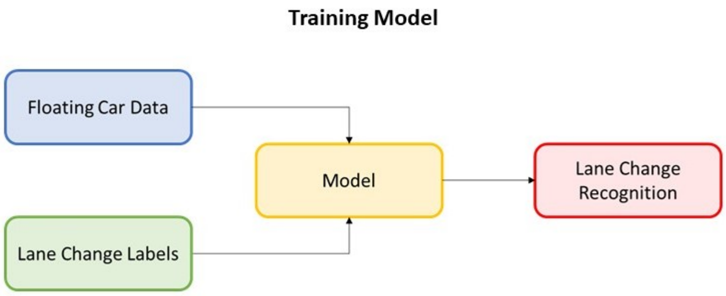


Figure 6.1: Schematic representation of model training

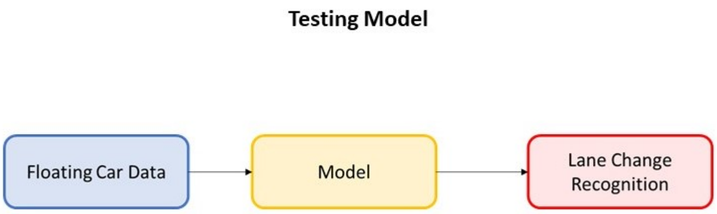


Figure 6.2: Schematic representation of model testing

6.2. Methodology and Data Preparation

After the matching- and reconstruction process of the trajectories as mentioned in section 4.4, it is known for each trajectory whether or not a lane change has been made in each road section in-between two loop detector locations. Since it is known from which to which lane a vehicle switches when making a lane change, several levels of detail can be deduced, from whether or not a lane change is made, to from which to which lane a vehicle moves. Each of these levels of detail at which a lane change can be labelled, can affect the level of recognition by the algorithm. The different lane changes have been labelled in each of the following ways:

- *Lane change yes/no.* In this case the number of lanes changed, nor direction of the lane change are considered relevant. It is solely looked at whether or not a lane change is made. Labels: [1/0]
- *Lane change to the left, no lane change, lane change to the right*. In this case all lane changes are labelled as a change in their corresponding direction regardless of the number of lanes that are changed in one go. Labels: [2/0/1]
- *Lane change to the left vs no lane change and lane change to the right.* In this case a lane change towards the left, regardless of number of lanes changed, is labelled as a category, and no lane change and lane(s) change(d) to the right is labelled together as another category. Labels: [2/0]
- *Lane change to the right vs no lane change and lane change to the left.* In this case a lane change towards the right, regardless of number of lanes changed, is labelled as a category, and no lane change and lane(s) change(d) to the left is labelled together as another category. Labels:[1/0]

Using different labels is relevant considering the fact that in previous research it was found that for example a lane change to the left has slightly different characteristics than a lane change to the right. (Toledo and Zohar, 2007). Therefore, by using these different categories of labelling, it is explored whether different types of lane changes are recognised differently, or more accurately than others, and whether different lane changes are represented by different features.

The following data was deduced from the Floating Car Data and used as input features for the Random Forest algorithm. Each of these data categories is thereby investigated as indicator for lane change recognition:

- *X-distance.* This is the lateral distance between the trajectory point and the centerline of the corresponding road segment.
- *Y-distance.* This is the distance between the trajectory point and the beginning of the road section. This corresponds to the distance since the last passed loop detector.
- *Speed.* This is the speed of the vehicle at every point.

- *Timestamp.* This is the timestamp corresponding to each data point, expressed in Unix time (indicated as epoch time: the number of seconds that have passed since midnight January 1, 1970).
- *Heading.* The direction in which the vehicle is driving at every time step, expresses in degrees. This is deducted from the difference in location between every consecutive data point. The exact calculation can be found in appendix A.5.
- *Heading difference to previous point.* This is the difference in heading of the vehicle compared to its heading at the previous data point. This heading difference is expressed in degrees.
- *Heading difference to the centerline.* This is the difference in heading of the vehicle, for each data point, compared to the direction of the centerline of the road segment at that point. By using the difference in heading, the direction of the infrastructure is taken into account, thereby correcting for any angles in the road. This measure therefore indicates more clearly the individual driving direction of the vehicle, rather than showing the direction of the road that is followed.

By transforming the X- and Y- coordinates of the vehicles to the X- and Y- distance between the vehicle and the road centerline (as X-distance) or the start of the segment (Y-distance), it is ensured that the model is not trained for a specific location, but rather can be used for any road section.

Using all above mentioned data, the different data sets for the random forest algorithm are prepared by firstly merging all the separate days of Floating Car Data into one set, which is limited to each homogeneous piece of road between two consecutive loop detector locations. For these road sections between the loop locations, a label is created indicating the type of lane change for each vehicle trajectory that has been reconstructed.
A visual overview is given in figure 6.3. This image shows an example of how road sections and lane changes are labelled. The vehicle is depicted on the loop in the lane at which it is detected at each loop detector location. A road section is the area between two consecutive loop detector locations. The label a vehicle gets is dependent on the level of detail of labelling, as well as the direction of the lane change. In the example image below, the vehicle trajectory is labelled as lane change YES for both road sections 2 and 3 in case of labelling lane changes as Yes/No. Or in case of indicating the direction of the lane change, the trajectory is labelled as lane change to left in road section 2, and lane change to the right in road section 3.

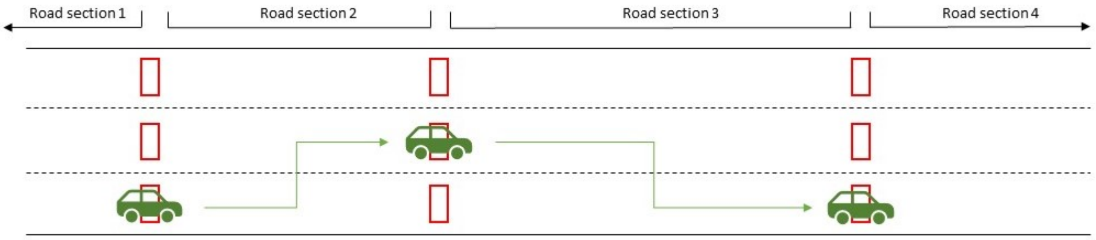


Figure 6.3: Example of road sections and labels

6.2.1. Equal number of data points

The Flitsmeister Floating Car Data consists of a data point per second, meaning that not all trajectories in a selected road section contain the same number of data points (as speed varies per vehicle). Furthermore, not all data points are spaced out equally due to changes in vehicle speed throughout a section. Since the input for a random forest algorithm requires a data set with equal number of features, it needs to be ensured that each vehicle trajectory consists of the same number of data points. In order to attain this, the vehicle with the highest number of data points (after filtering) within a section is determined. Each vehicle with a lower number of points within the same section is then augmented to reach the same number of points as that vehicle's trajectory. This is achieved by adding as many points as needed to the trajectory.

Each of the newly added points receives a value per feature which is the average of the value of the point before and after it of the same feature. These extra points are placed in such a way that they are spread out as equally as possible throughout the trajectory.

In figure 6.4 an illustrative example of the process of adding data points to the Floating Car Data is depicted. In the first image two trajectories with different numbers of data points are depicted. In the second image the trajectory of vehicle 2 is augmented with extra data points in order for it to match the number of data points of the first vehicle. These extra points are inserted at equally spaced locations, indicated by a light green point and an asterisk. Finally, in the last image the final trajectories of both vehicles consisting of equal number of data points are indicated.

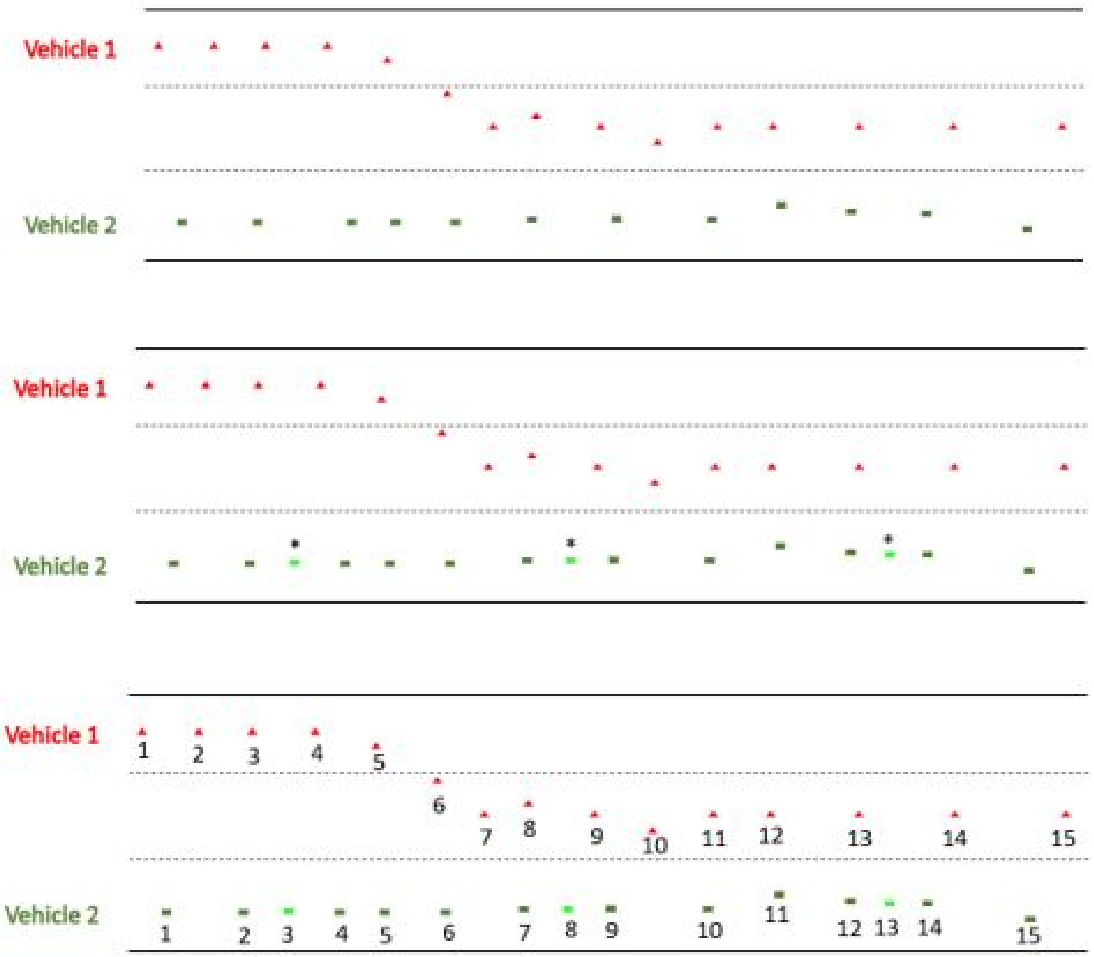


Figure 6.4: Example of adding extra data points

6.2.2. Data set balancing

As mentioned before, previous studies found that vehicles make on average 0.4 to 0.5 lane changes per kilometre, meaning that on road sections shorter than 1 kilometre a majority of vehicles do not make a lane change. Therefore, the data set of a road section contains more trajectories not making a lane change than those making a lane change within the section.

To ensure the model is trained on recognising patterns seen in the features, rather than attributing a higher chance of occurrence to the type of lane change appearing more often within the road section, the data set is balanced in such a way to consist of an equal number of trajectories for each lane change type (no lane change being considered a type of lane change).

This means that depending on the labelling of the data set (e.g. yes/no or left/ no / right, etc.), the data set will be filtered differently.
This filtering is done by splitting the data sets according to type of lane change (e.g. splitting the data set in a set with all the trajectories that make a lane change, and a set of all the trajectories not making a lane change). These split data sets are then resized in order for the sets to be the same size, by randomly removing trajectories from the data sets containing more trajectories. The now equally sized data sets of each lane change type are then merged together again leading to a data set with equal number of trajectories for each lane change type. Since the indexes of the original dataframe are kept, the order of the vehicles in the final data set is not dependent on the lane change type.

6.3. Normalisation and Analysis

As several features of the data have different units, with a different scale of values (e.g. speed in the hundreds versus heading mostly between -20 and 20), analysis comparing the different features can be offset due to these differences in units and size ranges. In order to be able to compare the different features, all data is normalised in such a way to have values between 0 and 1. Each driving characteristic (eg. speed, heading, etc..) is normalised individually. Firstly, the absolute value of all data is taken, after which the highest value of that specific driving characteristic is assigned 1 and the lowest 0, with all in-between values scaled according to the following formula:

$$normalised\_value = \frac{original\_value - smallest\_value}{maximum\_value - minimum\_value}$$

By looking more closely at the data and features of vehicles, it can be looked at whether differences are found between characteristics of vehicles making a lane change and those not making a lane change. In order to do so, all features (speed, heading, heading difference to the centerline, heading difference compared to the previous data point, timestamp, y-distance, and x-distance) are looked into separately, analysing statistics for each of them. For each of these features the normalised data set is split into yes lane change, and no lane change. Then, for each vehicle the sum of the normalised values per feature are taken, after which both the mean and the standard deviation over all vehicles are analysed. The results can be found in table 6.1 below.

	Speed	Heading	Heading difference to CL	Heading difference from previous data point	Timestamp	Y-distance	X-distance
Mean of sum - Yes LC	12.00	7.671	0.684	0.284	11.542	10.847	3.159
Mean of sum - No LC	12.00	7.660	0.681	0.278	11.517	10.823	3.150
Std of sum - Yes LC	1.6957	0.1936	0.3161	0.3628	6.7348	2.467	1.977
Std of sum - No LC	1.6671	0.1823	0.3163	0.3531	6.7122	2.464	1.985

Table 6.1: Mean and standard deviation of the sum of the absolute values of the data, for lane changes and no lane changes.

Since studies have found that the steering angle of lane change manoeuvres resembles a sine wave (Yoshida et al., 2008), and looking at the findings from Van Ballegooijen (2019), an example of a difference that might be expected in this analysis is that the sum of the absolute heading compared to the centerline of the road of a vehicle making a lane change would be larger than a vehicle not making a lane change.

From the results in table 6.1, it can however be seen that there are only very slight differences in values between cases in which lane changes are made and those in which no lane change is made. For all features except for speed, the values of vehicles not making a lane change are slightly lower than those that do make a lane change, but the difference is very small.
In order to analyse whether or not the difference in mean between the vehicles making a lane change and those not making a lane change is significant, a t-test is executed. The results can be found in table 6.2 below.

	Speed	Heading	Heading difference to CL	Heading difference from previous data point	Timestamp	Y-distance	X-distance
t-value	-0.033	0.428	0.753	1.139	0.281	0.748	0.346
p-value	0.974	0.668	0.451	0.255	0.779	0.455	0.730

Table 6.2: Results of t-test

As can be seen the p-values for all features are (far) above 0.05, indicating that there is no significant difference between (the mean of) the data of vehicles making a lane change and those that do not make a lane change.

6.4. Model

Random forest models are models used for regression and classification, which make use of multiple decision trees, of which each uses a subset of features selected by bagging (bootstrap aggregation). The model decides the final prediction by taking either the average of all decision tree results for regression, or the majority of votes in case of classification. Furthermore, an advantage of random forest models is that they offer insights in variable importance, which indicates the features that play the most important parts in the prediction.

Considering the strong pattern recognition characteristics of a random forest model, as well as the fact that the possibility of over-fitting is extremely low, this type of machine learning model is very suitable for the purpose of lane change recognition with the data available in this study. Additionally, the choice of a random forest model is further strengthened by Das et al. (2020), who suggest using a random forest model in cases where only vehicle kinematics are available, as is the case in this study. In this case since lane change labels are considered as categories, classification is used.

6.4.1. Evaluation method

There are several measures through which to evaluate the results and performance of a random forest model, of which the accuracy score and confusion matrix are commonly used and insightful, and together with the precision, recall, and F1-score offer a good overview of the model performance.

In order to measure how well the model is able to predict lane changes from the data, the first indicator to look at is the accuracy score of the test-data, which measures how many labels were predicted correctly by the model, compared to the total number of predictions made.

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{total predictions}}$$

The confusion matrix then offers additional information by indicating the distribution of the predictions made compared to the true labels of the data. This thereby demonstrates how many labels were predicted correctly and wrongly, as well as what kind of mistakes were made.

Finally, the precision, recall, and F1-score offer additional insight into the model performance by showing what type of errors the model made. The precision score indicates the level at which the model avoids false labelling (so false positives or false negatives) for each label. This is calculated as

$$\text{precision} = \frac{\text{true positives}}{\text{true positives + false positives}}$$

and can also be seen as the percentage of predictions which were correct for each label.

The recall value indicates the level of how accurately positive labels are found in each category, calculated as

$$\text{recall} = \frac{\text{true positives}}{\text{true positives + false negatives}}$$

Finally, the f1-score looks at the combination of the precision and recall matrices, indicating which percentage of positive predictions were correct. It is calculated as follows:

$$\text{F1 score} = 2 * \frac{\text{recall} * \text{precision}}{\text{recall} + \text{precision}}$$

.
By looking at the combination of these five evaluation metrics, the performance of the model can be evaluated.

6.4.2. Hyperparameter Tuning

When running a random forest model, there are several parameters which need to be set by the researcher, namely the hyperparameters. In order to find the most suitable hyperparameter values for the model, a randomsearch is executed, which aids in searching for the best values for each hyperparameter in the model. When using a randomsearch, multiple combinations of hyperparameters are tried and it is looked at how each combination impacts the accuracy of the model. In order to decide which values of hyperparameters are tested, a grid of ranges for the hyperparameters is decided on, after which a specified number of samples are randomly taken from this grid.
The following hyperparameters were evaluated in this research:

- n_estimators: This is the number of trees considered in the model. In general, more trees increase the performance of the model. However, more trees also increase the computational time of the model. Usually the model performance increases with an increasing number of trees, until a point where the additional trees do not increase accuracy significantly more, while the processing time of additional trees does increase. This is then considered a suitable number of trees. For the random search, the values of this hyperparameter to be considered were decided as 10 values between 10 and 1000, leading to the following: *n_estimators = [10, 120, 230, 340, 450, 560, 670, 780, 890, 1000]*

- min_sample_split: This hyperparameter indicates the minimum required observation in a node in order for it to split. The values considered for this hyperparamter are the following: *min_sample_split = [2, 5, 10, 20]*

- min_sample_leaf: Similarly to min_sample_split, this parameter controls the minimum samples in a node, however in this case it looks at the minimum number of samples required in a leaf node after splitting a node. In this study the values considered for this are: *min_sample_leaf = [1, 2, 4, 8]*

- max_features: This hyperparameter considers the maximum number of features which can be used per tree. The two options for this hyperparameter considered are ”Auto”, and ’sqrt”. ”Auto” indicates there is no restriction on the maximum number of features considered for a tree, while ”sqrt” takes the square root of the total number of features in an individual tree. *max_features = ’auto’, ’sqrt’*

- max_depth: With this hyperparameter the maximum depth a decision tree can grow is defined, which basically indicates the maximum number of splits to be made per branch. Often increasing the depth of a tree will increase the training result of the model, but not the testing result, since the model starts overfitting the training set. The random search considers 10 values between 5 and 110, leading to the following values for the maximum depth: *max_depth = [5, 16, 28, 40, 51, 63, 75, 86, 98, 110]*

- Bootstrap: Finally, the last hyperparameter which is tuned is whether or not to use bootstrapping in the model. When bootstrapping is used in a random forest, for each tree, the features used are randomly selected with replacement, meaning some features can be selected multiple times for the same tree. Bootstrapping is a method which helps to avoid over-fitting of a model. The random search therefore considers *bootstrap = ’True’, ’False’*

Considering the processing time of the random search, the number of iterations is set to 10, with 3 cross validations.

6.4.3. Results

From the data sets available, four different models will be fitted, one for each type of lane change labelling. For each of these models, first a base model is run, after which the hyperparameter tuning is executed in order to improve the initial model. For each model the most suitable hyperparameters will be indicated, as well as the level of improvement achieved by the hyperparameter tuning compared to the base model.

For each model both the training- and testing accuracy are indicated, as well as the confusion matrix and evaluation metrics in order to further analyse the model performance.

All models are run on data sets consisting of the following features: speed, x-distance, y-distance, heading, heading difference to previous data point, and heading difference to the centerline. The data sets used to fit the models are from location 1, for data from the 26th of June until the 2nd of July.
In order to test the accuracy of the models after fitting, they are all run on a data set with trajectories from the same road section from the 3rd and 4th of July. By testing the model on this unbalanced data set, it is found how the model would perform on data in proportions as found in traffic.

Model 1 (Lane change Yes/No):
The first model is fit using a data set labelled with lane change yes or no. An initial model is fit, leading to a training accuracy of 100%, but a testing accuracy of only 59.08%. Considering this high training accuracy, but much lower testing accuracy, the model is most likely overfit.

It is therefore important to tune the hyperparameters of the model through a random search. The most suitable hyperparameters for this model according to this random search were found to be:

- n_estimators: 670
- min_samples_split: 10
- min_samples_leaf: 8
- max_features: 'auto'
- max_depth: 16
- bootstrap: 'True'

The model was then run with these hyperparameters, which led to the following results:

Training accuracy = 89.80%
Testing accuracy = 60.61%

Clearly the training accuracy has reduced while the testing accuracy has increased, indicating the model is no longer overfit, and more accurately predicts the different types of lane changes. The tuning of the hyperparameters has for this model resulted in an improvement of the model accuracy of 2.59%.

From this tuned model, the following distribution of the labels was found, as shown in the confusion matrix below:

True label No LC	1797	1078
True label Yes LC	1187	1688
	Predicted label No LC	Predicted label Yes LC

Table 6.3: Confusion matrix tuned model

The wrongly predicted labels are found to be quite equally distributed between yes and no lane change, with no specific type of lane change being strongly over- or under-predicted. This indicates that the model is not biased to either of the two categories.
For further evaluation of the model, the following metrics as shown in Table 6.4 are assessed, of which the significance of each score has been presented in 6.4.1.

	Precision	Recall	f1-score	support
0	0.60	0.63	0.61	2875
1	0.61	0.59	0.60	2875
accuracy			0.61	5750
macro avg	0.61	0.61	0.61	5750
weighted avg	0.61	0.61	0.61	5750

Table 6.4: Evaluation model 1

The fact that the values for precision and recall are close together, indicates as well that the prediction for either label yes and no lane change is balanced quite equally.

In order to find out which features the model gave most value to in order to reach the predictions, the 10 most important features in the tuned model are extracted. For this first model these features are found to be:

- X-distance 22
- X-distance 21
- Heading 7
- X-distance 2
- Heading 14
- X-distance 20
- Heading 13
- Heading 8
- Heading 10
- Heading 9

The numbers here indicate the data point within the road section, in which 1 is the first data point in the driving direction, and 22 the last point within the road segment of location 1. For this model it can be seen that both the x-distance, representing the lateral distance between the vehicle and the centerline of the road section, as well as the heading of the vehicle are important predictors.

The model is then run on a different data set for validation. This data set contains trajectories of the 3rd and 4th of July from the same road section. The data set is not equally balanced, but contains the proportion of lane changes or no lane changes as found in traffic. A balanced accuracy of 62.02% is found from applying model 1 to this data set.

The following confusion- and evaluation metrics are :

True label No LC	2102	1016
True label Yes LC	1216	1568
	Predicted label No LC	Predicted label Yes LC

Table 6.5: Confusion matrix validation

	Precision	Recall	f1-score	support
0	0.63	0.67	0.65	3118
1	0.61	0.56	0.58	2784
accuracy			0.62	5902
macro avg	0.62	0.62	0.62	5902
weighted avg	0.62	0.62	0.62	5902

Table 6.6: Evaluation validation model 1

The accuracy found for this validation set is higher than the accuracy found for the test set of model 1. This can be explained by the fact that the data set of the validation data is not balanced, and so the label consisting of the larger group is found more often. This can also be seen from the precision and recall score, as well as from the f1-score which are all higher for category label 0 than 1, since category 0 is larger than 1.

Model 2 (Lane change Left/No/Right):
Next, a second model is fit for data of the same location and same days, but with labels for lane change to the left, no lane change, and lane change to the right. The same procedure is executed, in which first the base model is evaluated, after which the hyperparameters are tuned and the model is then run again.

In this case, the base model again reached a training accuracy of 100%, and reached a test accuracy of 47.90%. This 100% training accuracy together with a significantly lower testing accuracy indicates the model is overfit and needs tuning.
The random search executed for this model found the most suitable hyperparameters for this model to be:

- n_estimators: 670
- min_samples_split: 10
- min_samples_leaf: 8
- max_features: 'auto'
- max_depth: 16
- bootstrap: 'True'

These hyperparameters were then implemented for the model, which led to the following results for model 2 after tuning:

Training accuracy = 82.78%
Testing accuracy = 48.84%

Similarly as in model 1, the training accuracy reduced, while the testing accuracy increased. Overall, the hyperparameter tuning for this model resulted in an improvement of the model accuracy of 1.97%.

It is evident that the accuracy of this model is lower than that of model 1, which can be explained by the fact that this model categorises three types of lane changes, whereas the first model only had to distinguish between two categories.

Table 6.7 illustrates the distribution of the predicted and true labels of the tuned version of model 2:

True label LC Left	632	336	270
True label No LC	359	520	359
True label LC Right	294	282	662
	Predicted label LC Left	Predicted label No LC	Predicted label LC Right

Table 6.7: Confusion matrix base model 2

Similarly as in model 1, it can be seen that the distribution of the wrongly labelled lane changes is quite equal, indicating that none of the labels is specifically over- or under-predicted. Again this is confirmed by the precision, and recall score being close together, as can be seen in Table 6.8.

	Precision	Recall	f1-score	support
0	0.46	0.42	0.44	1238
1	0.51	0.53	0.52	1238
2	0.49	0.51	0.50	1238
accuracy			0.49	3714
macro avg	0.49	0.49	0.49	3714
weighted avg	0.49	0.49	0.49	3714

Table 6.8: Evaluation tuned model 2

Lastly, the 10 most important features indicating lane changes are found to again be X-distance and heading, as can be seen in the list below:

- X-distance 22
- X-distance 21
- X-distance 20
- X-distance 19
- Heading 10
- Heading 11
- Heading 18
- Heading 12
- X-distance 18
- Heading 8

For this second model as well, the model accuracy was tested on a validation data set from the same location for the 3rd and 4th of July. This led to a balanced accuracy of 50.89%, which is slightly higher than the accuracy found from the test set.

The following confusion matrix and evaluation scores indicate the distribution of false positives, negatives etc of the validation set:

True label LC Left	777	473	356
True label No LC	748	1502	868
True label LC Right	204	313	661
	Predicted label LC Left	Predicted label No LC	Predicted label LC Right

Table 6.9: Confusion matrix validation model 2

	Precision	Recall	f1-score	support
0	0.66	0.48	0.56	3118
1	0.35	0.56	0.43	1178
2	0.45	0.48	0.47	1606
accuracy			0.50	5902
macro avg	0.49	0.51	0.48	5902
weighted avg	0.54	0.50	0.51	5902

Table 6.10: Evaluation validation model 2

Model 3 (Lane changes Left):

The third model to be fitted, looks at recognising lane changes to the left. The data set used therefore contains only two labels, namely lane changes towards the left labelled as one category, and lane changes towards the right and no lane changes merged together into another category. Similarly to the previous two models, the data set on which this model is trained contains all features for location 1 for the 26th of June until the 2nd of July.

The base model for this case is once again found to overfit, with a training accuracy of 100%, but a testing accuracy of 62.63%.

A random search is therefore again executed, which found the following hyperparameters as being the most suitable for this specific model:

- n_estimators: 670
- min_samples_split: 10
- min_samples_leaf: 8
- max_features: 'auto'
- max_depth: 16
- bootstrap: 'True'

With the implementation of these hyperparameters, the following results were found from the tuned model:

Training accuracy = 91.58%
Testing accuracy = 63.98%

This indicates the hyperparameter tuning got rid of the overfitting, and led to an improvement of 1.07%.

The following confusion matrix (table 6.11) is related to this model, and once again shows the distribution of the predicted labels. Furthermore, the evaluation metrics are indicated in table 6.12, supporting findings in the confusion matrix:

True label LC Left	1038	599
True label LC Right + No LC	603	1035
	Predicted label LC Left	Predicted label LC Right + No LC

Table 6.11: Confusion matrix tuned model 3

	Precision	Recall	f1-score	support
0	0.63	0.63	0.63	1638
2	0.63	0.63	0.63	1637
accuracy			0.63	3275
macro avg	0.63	0.63	0.63	3275
weighted avg	0.63	0.63	0.63	3275

Table 6.12: Evaluation tuned model 3

From these evaluation scores of model 3, it is found that for both labels, as well as for all scores, a value of 0.63 is found. The fact that all values are found to be the same, can be explained firstly by the fact that the precision and recall rate are both the same. Considering the definition of the f1-score, this score automatically becomes the same as the precision and recall scores when those are equal.
The precision and recall score being equal, can be explained by the fact that an equal amount of false positives, and false negatives are predicted by the model.

The following 10 features are in this model found to be the most important indicators of the type of lane change:

- X-distance 22
- X-distance 21
- X-distance 20
- X-distance 19
- X-distance 18
- X-distance 17
- Heading_diff_CL 22
- X-distance 15
- Heading 19
- Heading 18

As in the previous two models, X-distance and heading are found to be indicators for lane changing. It should be noted however, that in this model, considering the fact that the X-distance is found more often than the heading, the X-distance is found to be more influential than the heading of the vehicle in lane change recognition, as compared to the first two models.

Similar as in the previous models, the model was then validated by implementing it on the data set of the 3rd and 4th of July, from which a balanced accuracy of 61.10% was found, along with the following distribution of true and predicted labels, and evaluation metrics:

True label LC Left	972	634
True label LC Right + No LC	1437	2859
	Predicted label LC Left	Predicted label LC Right + No LC

Table 6.13: Confusion matrix validation model 3

	Precision	Recall	f1-score	support
0	0.82	0.67	0.73	4296
2	0.40	0.61	0.48	1606
accuracy			0.65	5902
macro avg	0.61	0.64	0.61	5902
weighted avg	0.71	0.65	0.67	5902

Table 6.14: Evaluation validation model 3

From these evaluation metrics in Table 6.14, it can be seen that the amount of trajectories categorised as No LC or lane change to the right (label 0), is much higher than the number of trajectories changing lanes to the left. The precision of the lane changes to the left is also much lower than that of category 0.

Model 4 (Lane changes Right):
The last model which is trained is one looking at recognising lane changes to the right. Therefore, the data set, in a similar manner to the previous one, is labelled in two categories: lane change to the right as one category, and lane change to the left and no lane change together as another category. The results found for the overfitted base model is found to have a training accuracy of 100%, and a testing accuracy of 63.49%.

The random search is executed in order to tune the hyperparameters and resolve the overfitting. From this search, the following hyperparameters are found to be the most suitable for this model:

- n_estimators: 230
- min_samples_split: 5
- min_samples_leaf: 4
- max_features: 'sqrt'
- max_depth: 5
- bootstrap: 'False'

Implementation of these in the model lead to the following results for the tuned model:

Training accuracy = 70.05%
Testing accuracy = 64.50%

An improvement of 1.59% is thereby found as a result of the hyperparameter tuning.

The following confusion matrix goes along with this, as well as the evaluation metrics:

True label LC Right	749	489
True label LC Left + No LC	390	848
	Predicted label LC Right	Predicted label LC Left + No LC

Table 6.15: Confusion matrix tuned model 4

	Precision	Recall	f1-score	support
0	0.63	0.68	0.66	1238
1	0.66	0.61	0.63	1238
accuracy			0.64	2476
macro avg	0.65	0.64	0.64	2476
weighted avg	0.65	0.64	0.64	2476

Table 6.16: Evaluation tuned model 4

For this model again, the labels are found to be rather balanced, and the precision and recall scores are not too far apart, thereby indicating that the prediction of the labels is equivalent.

For this fourth model, the following features were found to be the 10 most important indicators of lane changes:

- Heading 10
- Heading 12
- Heading 9
- X-distance 22
- Heading 11
- Heading 17
- Heading 18
- Heading 14
- Heading 7
- X-distance 20

Again, both X-distance and heading of the vehicle were found to be predictors, however unlike model 3, in this case heading is found to be more significant than X-distance.

In order to have a comparison with the other three models, and to see the accuracy on an unbalanced data set, this model is also implemented on the validation data set of the 3rd and 4th of July. The following balanced accuracy score, and confusion matrix are found for this :

Balanced accuracy score = 60.26%

True label LC Right	730	448
True label LC Left + No LC	1501	3223
	Predicted label LC Right	Predicted label LC Left + No LC

Table 6.17: Confusion matrix validation model 4

Accompanied by the following evaluation metrics:

	Precision	Recall	f1-score	support
0	0.88	0.68	0.77	4724
1	0.33	0.62	0.43	1178
accuracy			0.67	5902
macro avg	0.60	0.65	0.60	5902
weighted avg	0.77	0.67	0.70	5902

Table 6.18: Evaluation validation model 4

The precision score of category 1 (lane change to the right) is here found to be rather low. Indicating that a lot of false positives (i.e. predicted labels of lane changes to the right, but true label is LC left or No LC) are returned.

Summary of model results:

An overview of the models and their accompanying testing and validation accuracies can be found in Table 6.19 below.

Model	Labels	Testing Accuracy	Validation Accuracy
1	Yes / No	60.61 %	62.02 %
2	Left / No / Right	48.84 %	50.89 %
3	Left / No + Right	63.98 %	61.10 %
4	Right / No + Left	64.50 %	60.26 %

Table 6.19: Summary of model performance

In order to take into account the imbalance in size of the lane change categories in the validation set, the balanced accuracy is taken for the validation sets for all the models. This is calculated as indicated in Appendix A.6. These results indicate that the models recognising lane changes in only one direction perform better on the test data than the models considering both directions.
A smaller difference is however found when looking at the performance on the validation data set, which was the exact same data set for all four models. The accuracy scores of all binary classification models lie much closer together, with the first model actually scoring the best accuracy-wise.

In order to analyse the models further, and consider which model is most suitable depending on the exact goal of the model it is interesting to look into the precision, recall, and f1-scores, which are summarised for the four models in tables 6.20 and 6.21.

Macro and Weighted average

	Precision	Recall	f1-score
Model 1 - Testing model	0.61	0.61	0.61
Model 2 - Testing model	0.49	0.49	0.49
Model 3 - Testing model	0.63	0.63	0.63
Model 4 - Testing model	0.65	0.64	0.64

Table 6.20: Averages Precision, Recall, and F1-score of testing models

Macro average

	Precision	Recall	f1-score
Model 1 - Validation model	0.62	0.62	0.62
Model 2 - Validation model	0.49	0.51	0.48
Model 3 - Validation model	0.61	0.64	0.61
Model 4 - Validation model	0.6	0.65	0.6

Weighted average

	Precision	Recall	f1-score
Model 1 - Validation model	0.62	0.62	0.62
Model 2 - Validation model	0.54	0.5	0.51
Model 3 - Validation model	0.71	0.65	0.67
Model 4 - Validation model	0.77	0.67	0.7

Table 6.21: Averages Precision, Recall, and F1-score of Validation models

6.4.4. Analysis of Results

The results from the four models presented above show that lane changes can be recognised from Floating Car Data on this road section with an accuracy between 48.84% and 64.50%. The recognition of lane changes to one specific side (either to the left or to the right) is found to offer the highest recognition rate on the test data, higher than considering both sides in one model.
A possible explanation for this is that, as further explained later, different characteristics are considered in lane changes to the right compared to those to the left (i.e. heading vs. x-distance).

By looking into the confusion matrices of all models, it can be seen that the wrongly predicted labels are rather balanced in all four models. Since the models use a balanced data set with equal numbers of labels per category for training and testing, the fact that the same order of magnitude of wrongly predicted labels are found in each category, indicates that the model does not over- or under-predict a certain label. This is also confirmed when looking at the precision and recall scores, which are similar for the different labels within the test data sets of a model. In a balanced data set, in case the values of precision and recall are completely equal, this indicates that the same number of false positives and false negatives are labelled by the model. This in turn means that the model is able to classify each type of lane change within a model equally well.

This is only the case for the third model, in which precision and recall have the exact same values for both labels. For the other models the precision and recall scores are similar, but not exactly the same, meaning that the labels are relative equally predicted, but not exactly equally.

When comparing the results of the four models, there are some differences between them which should be taken into account.
First of all, the fact that the first, third and fourth model all deal with binary classification, while the second model involves multi-class classification, means the comparison of the second model with the rest should be done with caution.
The second model for example reaches a lower accuracy score compared to the other three models, which can be explained by the fact that this model needs to identify patterns for three different types of lane changes, whereas in the other models a differentiation only needs to be found between two types of manoeuvres.

Secondly, although all four models are trained on the same overall data set, the exact (number of) trajectories used for the different models is slightly different considering the data sets are balanced per label, and thereby different (numbers of) trajectories are removed in the different models. The first model is for example trained on almost double the number of trajectories compared to the fourth model.

For this reason, the validation data set of the 3rd and 4th of July is used to compare the different models, as it uses the exact same data set for all models. Furthermore, since this data is not balanced according to lane change type, it represents a more realistic image of the actual traffic situation, namely with a majority of the trajectories not making a lane change.

Since the data set is not balanced, meaning certain types of lane changes occur more often than others, it is more important to not only look at the accuracy score, but also at the precision and recall values. In the first model it is seen that the values of these two metrics lie close to each other, with a difference of maximum 0.05 between them. In the second model however, the difference between the precision and recall values is much larger, which thereby also leads to lower f1-scores. In both the third and the fourth model a large difference is found between both the precision- and f1-scores for each label. This is caused by the fact that the category containing two types of lane changes (LC Right + No LC in model 3, and LC Left + No LC in model 4) is much larger than the category containing one type of lane change. For the validation of both these models, the precision score for the label of the smaller category is much lower than the recall score of this category. For the bigger category, the exact opposite is the case, where the precision score is much higher than the recall rate.

It is also found that the accuracy for the validation is higher for the first two models (lane change Yes/No, and lane change Left/No/Right), but lower for the last two models (lane change to the Left/No+Right, and lane change to the Right/No+Left).

In order to define which model is most suitable for lane change recognition, the question needs to be posed what the immediate goal of the lane change recognition is. For models with a higher precision score, the false positive rate will be lower, which is useful when wanting to know specifically the vehicles which made a lane change. A higher precision score thereby indicates the model is more certain the vehicle made the lane change if it is indicated to have done so. It however does not give information on the vehicles missed, which are not labelled as making the lane change when they actually did (the false negatives).
By looking at the recall rate, it is found how well the model recognised a type of lane change from all these type of lane changes made. The higher the recall rate, the less of the lane change types are missed by the model. When looking for a model which is able to recognise as many of a certain lane change type as possible, but is less sensitive to false positives, a model with a high recall rate would be suitable.
When looking for a model which takes both into account, not placing more value on one or the other, the model with the highest f1-score should be chosen.

Considering the fact that a random forest is a black box model, it is rather difficult to understand what the model has learnt as a rule to define a lane change. However, the model does offer insight into feature importance, which indicates how important each feature is in the classification. When looking at the feature importance of the different models, it can be seen that for all of them the x-distance (lateral distance of the vehicle) and the heading of the vehicle are important indicators. Considering the foundation of a lane change consists of a sinus-shaped movement involving a change in heading as well as a change in lateral position, this is a logical outcome.
An interesting finding is that in model 1 and 2 the importance of the X-distance and Heading are similar with both being represented almost equally in the top 10 most important features (40/60 in model 1, and 50/50 in model 2 for X-distance/Heading respectively). In model 3 on the other hand, X-distance is found to be more representative (70% X-distance/30% Heading), and in model 4 Heading is found more often in the top 10 features (20% X-distance/80% Heading).
Since model 3 looks at distinguishing lane changes to the left, and model 4 looks into lane changes towards the right, this finding indicates a difference in lane change recognition to the left and the right. In which lane changes to the left are recognised by looking at the X-distance of the vehicle to the centerline of the road, while lane changes to the right are distinguished by looking at the heading of the vehicles. Considering model 1 and 2 look at lane changes in both directions, the finding of both X-distance, and heading being important in those models thereby also matches.

A possible explanation for this finding could be that when making a lane change to the left, the vehicle proceeds towards a faster lane, while with a lane change towards the right the vehicle changes to a slower driving lane. Generally speaking, vehicles are not able to accelerate at the same pace as they are able to reduce speed through braking. Lane changes towards the left requiring acceleration, and therefore involve more time and a smoother movement compared to lane changes to the right, which can involve more sudden steering movement (i.e. to merge in-between two vehicles). This can explain why the lane changes to the left are characterised by x-distance, while lane changes to the right are recognised through heading.

7 Discussion and Conclusion

This study has looked into the recognition of lane changes from Floating Car Data by looking at a rule-based method as well as by the implementation of random forest models. In this section the findings and their implications will be discussed. The research questions will then be answered, and a conclusion is drawn. Lastly, recommendations for future research are provided.

7.1. Discussion

7.1.1. Trajectory reconstruction

In order to label the trajectories in this study by the type of lane change, the trajectory reconstruction algorithm by Arman and Tampere (2021) was applied, which reconstructs the Floating Car trajectories according to loop detector matching. Through this implementation, the labels and thereby the ground truth were created.
From this reconstruction it became clear that some trajectories require more severe corrections than others, which indicates the GPS error is not the same in all road sections and devices.

It must be considered that, although in this study the labels found by the reconstruction algorithm are regarded as ground truth, this method is sensitive to errors and a 100% accurate labelling can not be guaranteed. Especially considering the fact that this is the first time the method is applied outside of the original study environment, on a different road section, and even a different country.
Furthermore, when applying the method, manual placement of the loop detectors and road placement was required. Although this was done with caution, an error is easily made in such a process, which could have influenced the matching. Especially for the placement of the loop detectors, the passing time of the vehicle over the loop is second specific, meaning a slightly wrong placement of the detector can influence the matching.

The original study in which this algorithm is presented and tested uses multiple methods for validation, such as drone images and d-GPS. Since this study does not use any of those methods to check the accuracy of the lane changes found, it is unknown whether the matching has been done equally successfully. This is a limitation of the study, which needs to be taken into account when drawing conclusions.

Another reason to critically assess the ground truth used, is the fact that 1.8 to 3 times more lane changes are found in the case study area compared to what would be expected according to literature. This might seem to be a large number, but might be explained by the fact that the road sections on which the analysis is done is located at, and around, a weaving section, leading to more lane changes. Furthermore, the study from which the average number of lane changes was deduced mentions that its results might be site-specific.

A final reason to consider the ground truth with caution, is that due to human error a number of loop detectors present in the research area were not considered for the matching. This does not necessarily lead to wrong results seeing that no false information is used, but it can influence the accuracy of the matching process. The more loop detectors considered in the matching process, the more information, and therefore a reduced probability of incorrect matching.

7.1.2. Rule-based method

After the process of trajectory reconstruction, a rule-based method of lane change recognition using the heading difference of a vehicle with the road infrastructure, as developed by Van Ballegooijen (2019), was assessed. When applied to this study's research area on the A27 with the associated available data, a rather low number of lane changes was found. Especially when comparing to the number of lane changes expected according to literature, and the reconstructed trajectories.

Combining this result with the findings in Chapter 6, it could be reasoned that since the rule-based method looks at delta heading, and heading is found to be an important indicator for lane changes to the right, perhaps the rule-based method has a low recognition rate due to only recognising lane changes to the right.

In order to verify whether this was indeed the case, the percentage of lane changes recognised per direction by the rule-based method was investigated. No significant difference was however found between the recognition rates of lane changes to the left compared to those to the right. The reason for the low recognition rate by the rule-based method is therefore not caused by this factor.

7.1.3. Random Forest algorithm

The next part of this research looked into the use of machine learning, and more specifically random forests for lane change recognition. The labels used in this model originate from the ground truth found through the loop detector matching.

The four models indicate the level at which lane changes can be recognised according to different types of labelling. The three binary-classification models (models 1, 3 and 4) result in higher recognition rates compared to the multi-class classification model (model 2). As mentioned before, from a statistical point of view, this is an expected finding, considering the fact that more categories also come with more possibilities of false positives and negatives.

From a model point of view, this finding is however remarkable. The fact that models 3 and 4, which look into lane changes in a single direction, score remarkably higher than model 2, raises questions about the difference in lane changing patterns per direction. If a clear difference can be found between lane changes in a specific direction as seen in models 3 and 4, it would be expected that this difference can also be recognised when differentiating between the three directions in one go. Especially taking into account that the size of each category in model 2 is larger than those in model 3, and equal to those in model 4, indicating that reduced training size is not the cause of the lower accuracy.

This issue indicates clearly one of the downsides of a black-box model, namely that it is nearly impossible to fully discover which rules a model has learnt in order to classify the data, apart from looking into the feature importance.

In order to find out the extent to which a rule can be deduced from a random forest algorithm, an experiment was done in which a model was fit on the data set labelled according to the rule-based method from chapter 5. By doing this, the exact rule by which the data set was labelled was known by the researchers. After fitting the random forest model on this data, it was aimed to extract whether the model had learnt the same rules in order to achieve the labelling. This was found not to be the case, which proves the difficulty of a black box model such as random forests.

The analysis of the random forest fit on the rule-based method can be found in appendix A.7

For all of the four models trained, the base model overfit on the data set, which indicates the importance of hyperparameter tuning. By tuning the hyperparameters of the models, the testing accuracy was improved in all cases. The random search which was executed to achieve this made use of 10 iterations with 3 cross validations. Furthermore, a set number of values for each hyperparameter were picked, as indicated in section 6.4.2. All of these decisions impact the tuning of the models, and it is possible that with a wider choice in values, and a higher number of iterations, hyperparameters would have been tuned slightly differently, possibly leading to further increased accuracy. Ideally, in order to achieve the

most ideal hyperparameter tuning, a grid search can be executed, which looks into every combination of hyperparameter options, rather than a random sample of combinations as done with a random search. A grid search however is extremely demanding in both time and computational power, which were not available during this research.
In case future research will progress with this study, it could be interesting to look into further tweaking of the hyperparameters, and how much the accuracy of the models would thereby improve.

All four models presented offer their own value in lane change recognition, depending on the level of detail of a lane change that needs to be found. Although, as mentioned before, the four models cannot be compared one-on-one due to different types of classifications and different data set sizes used for training, a certain level of comparison can be reached by looking at the validation of the models. Since the validation is performed on the exact same data set for all four models, the difference between their findings can be used to decide the model that is the most suitable for the goal it is chosen for.

The goal of recognising lane changes from Floating Car Data is ultimately to be able to offer personalised, in-car driving advice to drivers. The most suitable model for this case therefore depends on the exact application of this advice.
For purposes in which it is crucial that a lane change is accurately indicated (for example for safety reasons), a model with high precision score is needed. On the other hand, in cases where as many drivers as possible making a certain lane change need to be reached, in which case it is not critical if a few drivers are wrongly contacted, a model with a high recall score is more suitable. In case both are equally important, a model with high f1-score and accuracy should be used.
Of course also depending on whether information on the direction of a lane change is required, or whether it is sufficient to know whether or not a lane change is made, some models can be more suitable than others.

Considering the fact that both the study by Van Ballegooijen (2019) and this study find a maximum recognition of lane changes of around 50% to 64%, it seems that with the current level of GPS accuracy, and the corresponding precision of Floating Car Data, a higher level of lane change recognition might not be achievable from only Floating Car Data, at least by means of this method. By improvement of the GPS receivers in mobile devices and navigation systems, a higher accuracy can probably be reached in the future.

Das et al. (2020) in their study did however find a higher lane change recognition rate, which creates a suitable comparison in order to distinguish the differences between the studies, and thereby focus points for future research. A major finding by Das et al. (2020), which could explain the difference in results is the weather conditions, which were found to have a considerable effect on both the quality of the data, and the lane change behaviour. All vehicle kinematics were found to differ between weather conditions, and data collected during extreme harsh weather conditions were excluded for some models in the study. This is a parameter which is not taken into account in our study, and could impact the results. In case the weather during our study period was differing, it is possible that different lane change manoeuvres were made, or data quality differed per day. A second difference is the vehicle kinematics features used, namely the speed, longitudinal acceleration/deceleration, lateral acceleration/deceleration, and yaw rate, of which for each the mean, maximum, minimum, and standard deviation were used. This demonstrates a difference with the features used in our study, and thereby potentially leading to different results. Finally, another reason for the higher accuracy reached in the models by Das et al. (2020) could be the fact that a grid search was used for hyperparameter tuning. As mentioned earlier, such a method of tuning is more exhaustive and thereby leads to better tuning and results.

7.2. Conclusion

The main research question of this research was *'To what extent can lane changes be recognised from Floating Car Data?'*. In order to answer this question two sub-questions were looked into, namely *'What (level of) information can be obtained from Floating Car Data?'*, and *'Which trajectory characteristics are significant in lane change recognition?'*. By looking into these questions the main research question can then be answered.

The first sub-question, *'What (level of) information can be obtained from Floating Car Data?'*, has been looked into mainly in chapter 4.
Raw Floating Car Data retrieved from in-car navigation systems and mobile navigation apps offer vehicle trajectory information at a 1 HZ frequency. The data consists of a timestamp, GPS location, and speed of the vehicle for every second.
In The Netherlands an often used navigation app collecting Floating Car Data is Flitsmeister. This service is used by approximately 1.8 million users in The Netherlands, thereby covering a substantial amount of road users. The Floating Car Data used in this research consists of Flitsmeister data.

From the analysis of the raw Floating Car Data, it was found that the GPS location indicating the vehicle position at a 1 HZ frequency is often inaccurate. Not only is a large number of the data found to lie outside of the driving lanes, trajectories are also found to make strong zig-zagging movements which are often unrealistic for vehicles to make. From this it was concluded that GPS errors are present in the data, and lateral movements found in the GPS positions of the vehicle can therefore not simply be regarded as a lane change.

The second sub-question, *'Which trajectory characteristics are significant in lane change recognition?'*, is mainly answered in chapters 5 and 6.
Using a rule-based method looking at the delta heading of vehicles at consecutive time steps has, in this study, been found to recognise only a very limited amount of lane changes.
However, when using a random forest model, lane changes can be more accurately recognised through the heading of the vehicle, and the lateral distance between the vehicle and the centerline of the road (X-distance).
More specifically, lane changes to the right are signified by the heading of the vehicle, and lane changes to the left are identified by the x-distance between the vehicle and the centerline. When looking into lane changes independent of the direction, a combination of the heading and x-distance of the vehicle are found to be indicators.

From these lane change recognition models created through a random forest algorithm, it is found that lane changes can be recognised with an accuracy of more than 60%, with higher recognition rates reached using binary classifications than multi-class classification.

The overall research question of this research *'To what extent can lane changes be recognised from Floating Car Data?'* is thereby answered, and a model has been created with which it has become possible to recognise lane changes from Floating Car Data with an accuracy of up to 64%. This accuracy level lies in the same range as the results found by Van Ballegooijen (2019), and indicates that at present, and with the current GPS sensitivity, a higher accuracy of lane change recognition can not be reached from Floating Car Data, at least through these methods.

The most suitable model out of the four models created is dependent on the exact requirements and type of lane change aimed at being recognised.

7.3. Recommendations for future research

Considering the findings from this research, a number of interesting future research directions have established.

First of all, it would be valuable to implement the same research on a different road section. This would help determine whether the results found in this research are specific to this road section, or whether they are indeed generalizable for all lane changes on (Dutch) highways. Considering the completeness and unique composition of the data set used in this research, it is valuable to use this same set for further analysis. Within this data set there are numerous road sections for which lane change recognition models have not yet been created. It would also be valuable to merge the data of several road sections in order for a model to be trained on different road sections together.

On top of this, it would be valuable to look into a completely different road section of a different highway, with a different road orientation (e.g. East to West) to find out whether the findings are similar. Also taking into consideration the weather conditions can lead to higher recognition rates and is interesting to take into account in future research.

Another adaptation of the current research which would be valuable, is to reduce the filtering of the trajectories in the pre-processing of the data. This would offer a data set closer to the raw Floating Car Data, and thereby lead to easier implementation and wider use.

Finally, it would be interesting for future research to add a third data source such as a camera to a same type of research. This third source would offer an additional manner in which to deduct the ground truth, thereby increasing the reliability of the lane change labelling, and thereby lead to more reliable results.

7.4. Recommendations for practice

Practical recommendations found by this research are the improvement of GPS receivers in mobile devices and navigation devices. This would lead to reduced GPS error, and thereby more reliable trajectory data, which would likely improve the model's accuracy.

Bibliography

ANWB. (2020). 17 procent meer files op de nederlandse wegen in 2019. https://www.anwb.nl/verkeer/nieuws/nederland/2019/december/knelpunten-2019

Arman, M., & Tampere, C. J. (2021). Lane-level trajectory reconstruction based on data-fusion. version 1.0.2. *KU Leuven Working Papers in Industrial Management Traffic and Infrastructure (CIB)*.

Bullock, R. (2007). Great circle distances and bearings between two locations. *MDT, June*, *5*.

Cao, X., Young, W., & Sarvi, M. (2013). Exploring duration of lane change execution. *Australasian Transport Research Forum*.

Das, A., Khan, M. N., & Ahmed, M. M. (2020). Detecting lane change maneuvers using shrp2 naturalistic driving data: A comparative study machine learning techniques. *Accident Analysis & Prevention*, *142*, 105578.

Dogan, Ü., Edelbrunner, J., & Iossifidis, I. (2011). Autonomous driving: A comparison of machine learning techniques by means of the prediction of lane change behavior. *2011 IEEE international conference on robotics and biomimetics*, 1837–1843.

Floating car data via flitsmeister. (n.d.). https://www.flitsmeister.nl/fcd

Goodall, C., Syed, Z., & El-Sheimy, N. (2006). Improving ins/gps navigation accuracy through compensation of kalman filter errors. *IEEE vehicular technology conference*, 1–5.

Knoop, V. L., Duret, A., Buisson, C., & Van Arem, B. (2010). Lane distribution of traffic near merging zones influence of variable speed limits. *13th International IEEE Conference on Intelligent Transportation Systems*, 485–490.

Knoop, V. L., Hoogendoorn, S., Shiomi, Y., & Buisson, C. (2012). Quantifying the number of lane changes in traffic: Empirical analysis. *Transportation research record*, *2278*(1), 31–41.

Li, L., Lv, C., Cao, D., & Zhang, J. (2017). Retrieving common discretionary lane changing characteristics from trajectories. *IEEE Transactions on Vehicular Technology*, *67*(3), 2014–2024.

Louppe, G. (2014). Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502*.

Makridis, M., Mattas, K., Ciuffo, B., Raposo, M. A., Toledo, T., & Thiel, C. (2018). Connected and automated vehicles on a freeway scenario. effect on traffic congestion and network capacity. *7th Transport Research Arena*, 13.

Monot, N., Moreau, X., Benine-Neto, A., Rizzo, A., & Aioun, F. (2018). Comparison of rule-based and machine learning methods for lane change detection. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 198–203.

Roncoli, C., Bekiaris-Liberis, N., & Papageorgiou, M. (2017). Lane-changing feedback control for efficient lane assignment at motorway bottlenecks. *Transportation Research Record*, *2625*(1), 20–31.

Schlechtriemen, J., Wirthmueller, F., Wedel, A., Breuel, G., & Kuhnert, K.-D. (2015). When will it change the lane? a probabilistic regression approach for rarely occurring events. *2015 IEEE Intelligent Vehicles Symposium (IV)*, 1373–1379.

Skog, I., & Handel, P. (2009). In-car positioning and navigation technologies—a survey. *IEEE Transactions on Intelligent Transportation Systems*, *10*(1), 4–21.

Toledo, T., & Zohar, D. (2007). Modeling duration of lane changes. *Transportation Research Record*, *1999*(1), 71–78.

Treiber, M., Kesting, A., & Thiemann, C. (2008). How much does traffic congestion increase fuel consumption and emissions? applying a fuel consumption model to the ngsim trajectory data. *87th Annual Meeting of the Transportation Research Board, Washington, DC*, *71*, 1–18.

Van Ballegooijen, T. (2019). Rijstrookwisselingen weefvak a15 n3-sliedrecht west.

Wang, G., Sun, P., & Zhang, Y. (2019). Utilizing random forest and neural network to extract lane change events on shanghai highway. *Cictp 2019* (pp. 318–330).

Wu, N. (2006). Equilibrium of lane flow distribution on motorways. *Transportation research record*, *1965*(1), 48–59.

Yoshida, H., Shinohara, S., & Nagai, M. (2008). Lane change steering manoeuvre using model predictive control theory. *Vehicle System Dynamics*, *46*(S1), 669–681.

A

Appendix

A.1. Scientific Paper

Recognition of Lane Changes from Floating Car Data

Lotte Olthof, Victor Knoop, Joost de Winter, and Bart van Arem
e-mail: l.olthof-1@student.tudelft.nl

Abstract – **One of the current challenges withholding personalised lane-level driving advice is the inaccuracy and error of GPS signal from commonly used navigation devices and mobile phones. These GPS signals have an uncertainty margin up to several meters, therefore potentially indicating the vehicle location on a different lane than the actual lane it would be in. This unreliability therefore currently makes it impossible to accurately recognise lane changes from solely this data.**

This study looks into the recognition of lane changes from only Floating Car Data by the use of a Random Forest algorithm. In order to find the ground truth, a trajectory reconstruction algorithm is implemented, which uses the matching of trajectories with loop detector passages in order to find the lane a vehicle is in at each loop detector location. This information is then used to know whether, for each vehicle, a lane change is made on the road section in-between two consecutive loop detector locations. By training the model on this data, it was found that when using solely Floating Car Data, lane changes can be recognised with an accuracy of up to 64%. Indicators for lane change were found to be the lateral distance of a vehicle to the middle of the road, as well as the heading of the vehicle.

The study additionally looks into a rule based method of lane change recognition, which is compared with the Random Forest model.

Keywords – **Floating Car Data, GPS, Lane Change Recognition, Random Forest Model**

I. INTRODUCTION

The ability to recognise lane changes, or accurate lane-level location, from Floating Car Data, a data source of GPS traces from vehicles equipped with a navigation device or mobile phone navigation application, is an issue currently withholding personalised in-vehicle driving advice. Due to the inaccuracy of regular GPS, which can be offset up to several meters, it is not possible to say with certainty in which lane a vehicle is driving. For this reason, it is also not possible to deduct from the Floating Car Data alone, whether a vehicle makes a lane change, or whether the lateral movement seen in the data is cause by GPS error or distortion.

This study aims to fill this gap, and researches the possibility of recognising lane changes from Floating Car Data of free-flowing traffic on a Dutch Highway, by looking at both a rule-based method, and by training a Random Forest Model.

Dependent on the exact purpose of the lane change recognition models, and the desire of whether to recognise specific manoeuvres with high certainty, or whether the priority is to cover the highest number of lane changes as possible, certain models might be more suitable than others.

II. Literature Review on Lane Changes

Lane changes belong to some of the most frequent driving manoeuvres executed in free-flow traffic, with a vehicle making on average 0.4 to 0.5 lane changes per kilometre [1]. Both mandatory and discretionary lane changes are required for smooth driving experiences. The intensity of these manoeuvres is higher near network nodes and weaving sections, as well as in high density traffic situations [1] [2]. Definitions of lane changes and their associated beginning- and end-point vary among studies, ranging from the moment a vehicle's wheel crosses the lane boundary, to the moment the centre of the vehicle has reached the destination lane. Consequently, lane change duration is found to vary between 1 and 16 seconds, with an average of 5 to 6 seconds [3] [4].

GPS receivers tracking vehicle's location are present in numerous devices such as smartphones and in-car navigation. This offers the opportunity to track vehicle location and thereby certain aspects of driving behaviour. The location accuracy can however differ largely between trajectories due to in-vehicles causes such as receiver quality, internal filtering, or placement in the vehicle, as well as by external factors such as interference, atmospheric conditions, tunnels etc. [2]. This inaccuracy of GPS location due to data drift and errors can lead to an offset of several meters. In order to use Floating Car Data for lane change recognition, the exact GPS positions of a trajectory up to lane level can therefore not be relied on.

Multiple studies have been executed using additional data sources such as camera or steering wheel angle for lane change recognition. However, to the best of our knowledge, no scientific research has, to date, been done on lane change recognition from only Floating Car Data. An investigation has been carried out on lane change recognition from vehicle's delta heading (Δh), in degrees, deducted from Floating Car Data traces [5]. The delta heading represents the difference in heading between the vehicle and the road infrastructure, from which the following definition was found to recognise lane changes 67% of the time [5]. t_0, t_1 etc.. represent the time steps, per second, of a FCD point, and $\Delta h_{t_0}, \Delta h_{t_1}, etc$ represent the delta heading at each time step.

$$(\Delta h)\ at\ t_0,\ t_1,\ and\ t_2 < 0\ or\ (\Delta h)\ at\ t_0,\ t_1,\ and\ t_2 > 0$$

$$and$$

$$abs(\Delta h_{t_0} + \Delta h_{t_1} + \Delta h_{t_2} + \Delta h_{t_3}) >= 6$$

The rule was deduced by investigating trajectories which

This footnote will be used only by the Editor and Associate Editors. The edition in this area is not permitted to the authors. This footnote must not be removed while editing the manuscript.

were, according to the origin and destination road they were located on, known to have made at least one lane change. During validation on a different road section, the percentage of lane changes recognised was found to be between 50 and 70% depending on the road segment [5].

The increasingly growing field of research and application of machine learning in road traffic analysis offers new insights, including on lane change recognition. Numerous studies using this method focused on lane change recognition for autonomous vehicles, or looked into lane changing of vehicles relative to the positions of surrounding vehicles, which is not applicable for Floating Car Data. One study compared different machine learning algorithms such as Support Vector Machine, Artificial Neural Network, Random Forests, and eXtrem Gradient Boosting for lane change recognition using data of vehicle kinematics, machine vision, road characteristics, and driver demographics. It was found that eXtrem Gradient Booster led to the highest lane change detection accuracy of 95.9% [6]. When using only vehicle kinematics, the Random Forest model was found to score best, and the authors of the study thereby advice a Random Forest model for lane change recognition when using only vehicle kinematics (as is the case in Floating Car Data) [6].
Following this finding, along with the fact that research is lacking on lane change recognition from Floating Car Data as only source, this study aims to fill that research gap.

III. Method

In this study two data sources are used, namely Floating Car Data, and individual passage loop detector data. As a first step the trajectories from the Floating Car Data are filtered as described in section A.Both the data sources are then combined for trajectory reconstruction, as explained in section B.From this, lane changes can be deducted, from which labels are created which classify the trajectories accordingly. With the labelled data sets two methods of lane change recognition are applied, namely a rule-based method, and a machine learning algorithm. The rule-based method looks into the definition of a lane change as described in section II.and is described in section C. Finally, a random forest algorithm is implemented in order to train classification models for lane change recognition, as explained in section D.In section E.the case-study locations on which the research is implemented are described.

A. Filtering of data

For use in this study, the Floating Car Data is filtered according to the following definitions:

- Only trajectories within the study area (the A27 between Utrecht Noord and knp Eemnes) are considered, all parts of the trajectories outside the X- and Y- coordinates of the area of interest are excluded by manner of X- and Y-coordinate limiting. This implies removing all data points outside the study area by cutting of the trajectories at the edge of the section of interest. This ensures no information from within the study area is lost.
- Trajectories containing an unproportionally large number of data points, in which a large number of data points is defined as anything more than the number of data points corresponding with a vehicle driving 70km/h on the specified segment. By following this selection, both congestion, and the trajectories of vehicles which are present in the study area for an abnormal amount of time, e.g. those stopping at a fuel station, are filtered out.
- Trajectories containing a very small number of data points. The threshold for this amount is set as the number of data points that a vehicle driving 150km/h would have in the determined segment. By removing these trajectories, both incomplete trajectories, as well as trajectories of vehicles driving abnormally fast are filtered out.
- Only trajectories which are able to be reconstructed according to the trajectory reconstruction method are included. This process and requirements for such reconstruction is explained in further detail below.

B. Trajectory reconstruction

In order to achieve lane-level reliable trajectories from the imprecise Floating Car Data points, an algorithm is used through which the precise driving lane of a vehicle can be acquired at loop detector locations [2]. This trajectory reconstruction algorithm makes use of lane-level data fusion of Floating Car Data and individual passage loop detector data. The matching is executed by linking the passage time of a vehicle at the loop detector location according to the Floating Car Data, with the passage time registered by the loop detectors at the corresponding loop detector location. By this data-fusion it is known which loop the vehicle passes at every location, and thereby in which lane the vehicle is situated at those loop detector locations. A schematic overview can be found in figure 1.
The algorithm then reconstructs the trajectories between the loop detector locations, in order to create the more realistic vehicle trajectory. In order to execute this process, the original Floating Car Data trajectories are first filtered according to several requirements, as stated in [2].

In this current study, for which Floating Car Data and loop detector data of the A27 is available, the above mentioned algorithm is applied, thereby offering information on whether a lane change is made in each road section between two consecutive loop detector locations. Labels can be created accordingly, indicating the lane change, as well as direction, in a road section, for each trajectory in-between two consecutive loop detector locations.

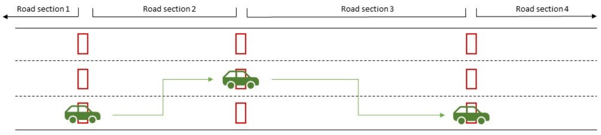


Fig. 1. Schematic overview loop detectors and road sections

C. Rule-based method

The lane change recognition rule as presented in section II.is applied to the data of this study, with the purpose of analysing the recognition of lane changes, as well as for comparison with the lane changes found by the trajectory reconstruction algorithm [5] [2]. The lane changes found by the rule-based method are compared with those found from the trajectory reconstruction algorithm, both in quantity and direction. Furthermore, the number of lane changes found is compared to the number of lane changes expected in the particular road section according to literature. Next, the rule-based lane change recognition method was slightly adapted to include an additional delta heading at an additional time step, thereby considering the delta heading at 4 time steps in the first part of the definition, and 5 time steps in the second part of the definition. The difference with the original definition, as well as the reconstructed trajectories and literature findings is then analysed.

D. Random Forest Model

A second method for lane change recognition is applied, with the use of machine learning, with which a model is trained by means of a random forest algorithm. The features are deducted from the Floating Car Data, and the labels from the lane changes are deduced from the vehicle location found through the trajectory reconstruction algorithm. Lane changes are labelled in four different ways, namely lane change yes/no, lane change to the left/no lane change/ lane change to the right, lane change to the left/ no lane change and lane change to the right, and lane change to the right/no lane change and lane change to the left.

Four different models are trained according to these different types of labelling. Each of these models is trained and tested on a balanced data set, in which the training and testing data is split with a 0.75/0.25 proportion. For each model a random search is executed in order to find the most suitable hyperparameters. After training, the models are also evaluated on a validation set, which consists of trajectory data of the 3rd and 4th of July from the same road segment. Since this data set is identical for all models, it is a good way in which to accurately compare the model performances. Furthermore, since this validation data set is not balanced, it consists of proportions of lane change types as found in traffic.

The models are evaluated according to their accuracy-, precision-, recall-, and f1-score, which are defined as follows:

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{total predictions}}$$

$$\text{precision} = \frac{\text{true positives}}{\text{true positives + false positives}}$$

$$\text{recall} = \frac{\text{true positives}}{\text{true positives + false negatives}}$$

$$\text{F1 score} = 2 * \frac{\text{recall} * \text{precision}}{\text{recall} + \text{precision}}$$

A high precision score indicates a low false positive rate, and a high recall score indicates a low false negative score. The f1-score is a balance between the two scores. Depending on the precise goal of the model a different indicator can be considered important.

E. Case-study locations

Two locations are decided on for which both the rule-based method and the random forest algorithm are applied. The first location, named location 1, is the road section between the two loop detector locations in segment 4, which has a distance between them of approximately 400 meters. The road section consists of 5 lanes, of which 2 are exit lanes in the second half of the section. Furthermore the most right-hand lane comes from a gas station. This ensures that lane changes will be made by vehicles that either enter the section from the gas station, or exit the road through the exit lanes. Lastly, the section is rather straight, which would reduce extreme GPS error due to road curvature and assure this section is a suitable road section on which to execute this study.

The second road segment, named location 2, is the road section between the first two loop detector locations of segment 5. This road segment has 3 lanes, all of which are through-lanes. The road section is approximately 500 meters long, and also quite straight. By choosing this second road section which does not contain entry- or exit lanes, a representative combination of types of road sections and accordingly, types of lane changes made is reached in combination with the first chosen road segment

IV. Data

This study is executed on data deduced from the Dutch highway A27, between Utrecht Noord and knp Eemnes, for which both Floating Car Data from Flitsmeister, and individual passage loop detector data are available. Within the research area trajectories of vehicles driving in Northward direction in the period between 26/06/2021 and 04/07/2021 are considered, with the exception of 30/06/2021, which was excluded due to abnormalities in that day's data set. The data is first analysed as indicated in sections Aand B.after which data sets are prepared for the different lane change recognition methods as explained in sections Cand D.Lastly, an analysis is done on the comparison of data from vehicles making a lane change, and those not making a lane change in section E.

A. Floating Car Data

The Floating Car Data comprises of trajectories of vehicles using the Flitsmeister navigation application. GPS location of the users is registered at a 1HZ frequency, as is the speed of the vehicle at each time step. All data is linked to an anonymous unique session ID per vehicle, of which around 45000 are registered per day.

Figure 2 depicts a 3-lane road from which the distribution of trajectories over the road can be seen. The horizontal axis represents the distance in meters from the centerline of the road, and the vertical axis indicates the position in meters in the spatial reference system. It can be seen that trajectories sway over and outside of the road as well as having oscillations between the lanes which are not realistic in human driving behaviour, thereby illustrating examples of GPS error found in Floating Car Data.

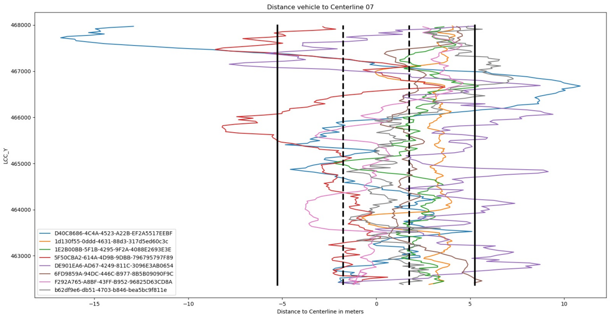


Fig. 2. Random trajectories on road section

In figure 3 additional analyses of the distribution of data points throughout the study area is demonstrated, from which it is concluded that a considerable number of data points are located outside of the driving lanes. The high number of data points located outside of the road in segment 9 is likely caused by the curvation of this road section, which is a known reason for increased GPS error. Segment 7 is excluded in this analysis due to high processing time caused by the length of said segment.

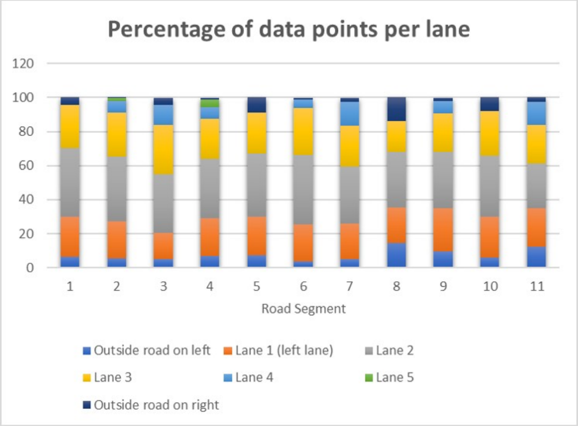


Fig. 3. Percentage of data points per lane

From the Floating Car Data it is possible to deduce additional trajectory features not directly found in the raw data. The following features are, for each data point, deduced from the FCD:

- *X-distance:* The lateral distance between the trajectory point and the centerline of the corresponding road segment.
- *Y-distance:* The distance between the trajectory point and the beginning of the road section. This corresponds to the distance since the last passed loop detector.
- *Speed.* This is the speed of the vehicle at every point.
- *Heading:* The direction in which the vehicle is driving at every time step, expresses in degrees. This is deducted from the difference in location between every consecutive data point.
- *Heading difference to previous point:* The difference in heading of the vehicle compared to its heading at the previous data point, expressed in degrees.
- *Heading difference to the centerline:* The difference in heading of the vehicle, for each data point, compared to the direction of infrastructure.

B. Loop detector data

Loop detectors are placed on a large number of highways throughout The Netherlands in order to measure traffic and traffic flow characteristics. With a passage over a loop, it is known what the speed, vehicle size, and time of passage over the specific loop is. The study area of this research consists of 33 loop detector locations, an overview of which is given in table I below.

Segment	1	2	3	4	5	6	7	8	9	10	11	12
Nr of loop detector locations	2	0	1	2	3	0	15	0	3	0	5	2

TABLE I

Number of loop detector locations per segment

Due to human error a number of loop detector locations are not considered in the study. The impact of this is however not considered significant.

C. Data set Rule-based method

The data used to analyse lane changes through the rule-based method as described in section III-Ç.consists of the heading difference between the vehicles and the centerline for all data points of trajectories remaining after filtering within locations 1 and 2.

D. Data set Random Forest Model

The implementation of a Random Forest Algorithm requires a data set with equal numbers of features for each trajectory. Considering vehicles drive at different speeds, while all generating data with a 1HZ frequency, each trajectory produces a different number of data points within a same road section.

To create the same number of data points for all trajectories within a segment, data points are added to the trajectories with lower numbers of data points than the trajectory with the highest number of points. Each of these added points has a value which is the average of the value of the point before and after it for that feature. The additional points are spaced out as equally as possible throughout the trajectory.

A second preparation required is the balancing of the lane change types in the data. This is necessary to ensure the model is trained according to patterns found in the data, rather than learning a higher chance of occurrence for a lane change type present more often within the data. The balancing is done by randomly removing a number of trajectories from the majority of the lane change type(s), ensuring all lane change types are represented an equal number of times within the data set. Through the balancing of the data four different data sets are created, one for each labelling type (Yes/No, Left/No/Right, Left/No and Right, Right/No and Left).

E. Analysis Lane Changes vs. No Lane Changes

Features of vehicles making lane changes can be compared to those not making lane changes to get a picture of any average differences in values between the manoeuvres. In order to analyse whether on average, differences can be seen between trajectories of vehicles making a lane change, and those not doing so, the normalised values of the absolute values of all features are compared. This ensures the direction of the movement is not considered, but only the fact that a difference occurs. In a road segment, for every feature, the sum of the normalised values of each trajectory are taken, after which the mean and standard deviation of these values is calculated for the trajectories making a lane change and those not doing so. The results of this comparison can be found in table II.

	Mean of sum Yes LC	Mean of sum No LC	Std of sum Yes LC	Std of sum No LC
Speed	12.00	12.00	1.6957	1.6671
Heading	7.671	7.660	0.1936	0.1823
Heading difference to Centerline	0.684	0.681	0.3161	0.3163
Heading difference from previous data point	0.284	0.278	0.3628	0.3531
Timestamp	11.542	11.517	6.7348	6.7122
Y-distance	10.847	10.823	2.467	2.464
X-distance	3.159	3.150	1.977	1.985

TABLE II

Mean and standard deviation of the sum of the absolute values of the data, for lane changes and no lane changes

To see whether the small differences in values found between the trajectories making a lane change and those not doing so are significant, a t-test is executed, of which the results can be found in table III. Considering the p-values are (far) above 0.05, no significant difference is found in values between vehicles making a lane change and those not making such manoeuvres for any feature.

	t-value	p-value
Speed	-0.033	0.974
Heading	0.428	0.668
Heading difference to Centerline	0.753	0.451
Heading difference from previous data point	1.139	0.255
Timestamp	0.281	0.779
Y-distance	0.748	0.455
X-distance	0.346	0.730

TABLE III

Results of t-test

V. Results

A. Rule-based method

Lane changes are recognised from vehicle's delta headings according to the definition indicated before. In its original study a recognition rate of 50-70% was reached, depending on the road section [5]. When applied to our study, a recognition rate of 53.10% and 51.98% was reached for location 1, and 2 respectively. However, as can be seen from the distributions in tables IV and V below, this method detects only around 7 to 8% of lane changes found by the trajectory reconstruction method using the loop detector matching.

Yes lane change Loop detector method	792	15597
No lane change Loop detector method	648	16367
	Yes lane change Delta heading method	**No lane change Delta heading method**

TABLE IV

Overview recognized lane changes according to different methods for location 1

Yes lane change Loop detector method	625	15495
No lane change Loop detector method	624	18127
	Yes lane change Delta heading method	**No lane change Delta heading method**

TABLE V

Overview recognized lane changes according to different methods for location 2

By comparison to the number of lane changes expected in these two road sections according to literature [1], it is found that only a small percentage of the lane changes is found by the rule-based method. The loop detector matching method however finds many more lane changes than expected, as can be seen in table VI. Comparing the number of lane changes recognised to the left and the right, it is found that approximately the same percentage is found by the rule-based method (4.1% to the left, 5.83 % to the right, and 3.4 % to the left, 4.79 % to the right for location 1 and 2 respectively), thereby indicating the cause of the lower recognition rate is not related to the direction of the lane change.

	Location 1	**Location 2**
Expected nr of Lane Changes from Literature	5344 - 6680	6974 - 8717
Percentage Lane Changes found by Rule-based method	21.6 - 26.9 %	14.3 - 17.9 %
Percentage Lane Changes found by Loop detector matching	245.3 - 307.7 %	184.9 - 231.1 %

TABLE VI

Number of expected and found lane changes

When applying the same analysis to the adapted definition, which considers the delta headings at an extra time step, similar results are found, indicating it is not a more suitable method than the original rule-based method.

B. Random Forest

Four models are trained for lane change recognition by use of the Random Forest algorithm, each of which uses different lane change labels.

Model 1 (Lane change Yes/No):
The first model is trained on the labelling of lane changes Yes/No, for which the following hyperparameter values were found to be most suitable:

- n_estimators: 670
- min_samples_split: 10
- min_samples_leaf: 8
- max_features: 'auto'
- max_depth: 16
- bootstrap: 'True'

The resulting confusion matrix of the model is seen in table VII.

	Predicted label No LC	Predicted label Yes LC
True label No LC	1797	1078
True label Yes LC	1187	1688

TABLE VII
Confusion matrix model 1

The trained model was additionally run on the validation data set, which led to the following result as shown in table VIII.

	Predicted label No LC	Predicted label Yes LC
True label No LC	2102	1016
True label Yes LC	1216	1568

TABLE VIII
Confusion matrix validation model 1

Model 2 (Lane change Left/No/Right):
From the random search executed for model 2, the same best hyperparameters were found as for model 1. The following results were found from the testing and validation model as found in tables IX and X respectively.

	Predicted label LC Left	Predicted label No LC	Predicted label LC Right
True label LC Left	632	336	270
True label No LC	359	520	359
True label LC Right	294	282	662

TABLE IX
Confusion matrix model 2

	Predicted label LC Left	Predicted label No LC	Predicted label LC Right
True label LC Left	777	473	356
True label No LC	748	1502	868
True label LC Right	204	313	661

TABLE X
Confusion matrix validation model 2

Model 3 (Lane changes Left):
For the third model again the same most suitable hyperparameter values were found as in models 1 and 2. The testing and validation results are found to be the following as seen in XI and XII:

	Predicted label LC Left	Predicted label LC Right + No LC
True label LC Left	1038	599
True label LC Right + No LC	603	1053

TABLE XI
Confusion matrix model 3

	Predicted label LC Left	Predicted label LC Right + No LC
True label LC Left	972	634
True label LC Right + No LC	1437	2859

TABLE XII
Confusion matrix validation model 3

Model 4 (Lane changes Right):
The last model was trained with different hyperparameters as found to be most suitable for the Random Forest algorithm, namely:

- n_estimators: 230
- min_samples_split: 5
- min_samples_leaf: 4
- max_features: 'sqrt'
- max_depth: 5
- bootstrap: 'False'

From which the distribution of predicted to true labels as indicated in XIII and XIV is found.

	Predicted label LC Right	Predicted label LC Left + No LC
True label LC Right	749	489
True label LC Left + No LC	390	848

TABLE XIII
Confusion matrix model 4

	Predicted label LC Right	Predicted label LC Left + No LC
True label LC Right	730	448
True label LC Left + No LC	1501	3223

TABLE XIV
Confusion matrix validation model 4

Summary of model results:
An overview of the models and their accompanying testing and validation accuracies can be found in Table XV below.

Model	Labels	Testing Accuracy	Validation Accuracy
1	Yes / No	60.61 %	62.02 %
2	Left / No / Right	48.84 %	50.89 %
3	Left / No + Right	63.98 %	61.10 %
4	Right / No + Left	64.50 %	60.26 %

TABLE XV
Summary of model performance

In order to take into account the imbalance in size of the lane change categories in the validation set, the balanced accuracy is taken for the validation sets for all the models.
From the results it can be seen that the confusion matrices are rather balanced in all four models, indicating no category is strongly over- or under- predicted. When comparing the model results, it should be considered that models 1, 3, and 4 are binary classification models, whereas model 2 is a multi-class classification model.
Random Forest models offer insight in which features are found to be most important in the classification process. In this study the most important features were found to be X-distance, and Heading, in which X-distance weighed more heavily in lane changes to the left, and Heading in lane changes to the right.

VI. Discussion & Conclusion

Considering the recognition rate of 50-70% for the rule based lane-change method in the original study, the findings in our study are unexpectedly lower. From the finding that heading is an important feature for lane changes to the right, and the rule-based method considers only delta heading, a potential explanation would be that lane changes to the left are not covered by this method. This has however been proven untrue since an almost equal percentage of lane changes to either side was found in the rule-based method.

The four models indicate the level at which lane changes can be recognised according to different types of labelling. The three binary-classification models (models 1, 3 and 4) result in higher recognition rates compared to the multi-class classification model (model 2). From a statistical point of view, this is an expected finding, considering the fact that more categories also come with more possibilities of false positives and negatives. From a model point of view, this finding is however remarkable. The fact that models 3 and 4, which look into lane changes in a single direction, score remarkably higher than model 2, raises questions about the difference in lane changing patterns per direction. If a clear difference can be found between lane changes in a specific direction as seen in models 3 and 4, it would be expected that this difference can also be recognised when differentiating between the three directions in one go. Especially taking into account that the size of each category in model 2 is larger than those in model 3, and equal to those in model 4, indicating that reduced training size is not the cause of the lower accuracy.

Deciding the most suitable lane change recognition model is dependent on the exact application of the lane change recognition. For purposes in which it is crucial that a lane change is accurately indicated (for example for safety reasons), a model with high precision score is needed. On the other hand, in cases where as many drivers as possible making a certain lane change need to be reached, in which case it is not critical if a few drivers are wrongly contacted, a model with a high recall score is more suitable. In case both are equally important, a model with high f1-score and accuracy should be used. Furthermore, depending on whether information on the direction of a lane change is required, or whether it is sufficient to know whether or not a lane change is made, some models can be more suitable than others.

Considering the fact that both the rule-based method and this study find a maximum recognition of lane changes of around 50% to 70%, it seems that with the current level of GPS accuracy, and the corresponding precision of Floating Car Data, a higher level of lane change recognition might not be achievable from only Floating Car Data, at least by means of this method. By improvement of the GPS receivers in mobile devices and navigation systems, a higher accuracy might be attainable in the future.

In our study, due to limited resources, a random search was executed for hyperparameter tuning. This however strongly limits the number of hyperparameters considered as compared to a grid search. In future research it would be interesting to implement a grid search for this study, which could increase the accuracy of the models.

Another addition which would strengthen the models is by applying this research to a different road section, and merging the data of several road sections together. This ensures the models are trained for more types of road sections, and therefore are more widely implementable.

Finally, it would be valuable to add a third data source such as a camera to a same type of research. This third source would offer an additional manner in which to deduct the ground truth, thereby increasing the reliability of the lane change labelling, and thereby leading to more reliable results.

REFERENCES

[1] V. L. Knoop, S. Hoogendoorn, Y. Shiomi, C. Buisson, “Quantifying the number of lane changes in traffic: Empirical analysis”, *Transportation research record*, vol. 2278, no. 1, pp. 31–41, 2012.

[2] M. A. Arman, C. M. Tampère, “Lane-level trajectory reconstruction based on data-fusion. version 1.0.2”, *102 KU Leuven Working Papers in Industrial Management Traffic and Infrastructure (CIB)*, 2021.

[3] X. Cao, W. Young, M. Sarvi, “Exploring duration of lane change execution”, *in Australasian Transport Research Forum*, 2013.

[4] T. Toledo, D. Zohar, “Modeling duration of lane changes”, *Transportation Research Record*, vol. 1999, no. 1, pp. 71–78, 2007.

[5] T. Van Ballegooijen, “Rijstrookwisselingen weefvak A15 N3-Sliedrecht West”, , 2019.

[6] A. Das, M. N. Khan, M. M. Ahmed, “Detecting lane change maneuvers using SHRP2 naturalistic driving data: A comparative study machine learning techniques”, *Accident Analysis & Prevention*, vol. 142, p. 105578, 2020.

A.2. Percentage of data points per lane

Road Segment	Outside road on left	1 (left lane)	2	3	4	5	Outside road on right
1	6.6 %	23.5%	40.3 %	25.3%			4.3%
2	5.4%	21.8%	38%	26%	6.7%	1.8%	0.3%
3	4.9%	15.6%	34.6%	28.9%	11.8%		3.9%
4	7%	21.9%	35.2%	23.4%	6.8%	4.7%	0.9%
5	7.4%	22.4%	37.2%	23.9%			9.1%
6	3.7%	21.7%	40.9%	27.6%	4.9%		1.1%
8	5%	20.7%	33.9%	24%	13.8%		2.5%
9	14.8%	20.8%	32.4%	18.2%			13.8%
10	9.8%	25.1%	32.9%	22.7%	7.2%		2%
11	5.8%	24%	36.1%	26%			8.1%
12	12.2%	22.6%	26.6%	22.6%	13.4%		2.6%

Table A.1: Percentage of data points per lane

A.3. Data Analysis Loop Detectors

Number of passages registered 26/06/2021	**Lane**				
Loop detector location	**1**	**2**	**3**	**4**	**5**
1	3755	15300	22601	34	
2	4212	15903	22159	5	
3	4079	15645	21950	2	
4	3659	13961	17467	4338	3249
5	3447	13301	18286	1511	5982
6	3684	13679	18299		
7	3743	13727	18150		
8	3601	13366	17739		
9	3914	14529	19556		
10	3950	14302	19015		
11	4003	14735	19266		
12	4213	14792	19018		
13	4383	14872	18720		
14	4410	14887	18485		
15	4371	14830	18800		
16	4316	14598	18346		
17	4491	15278	18199		
18	5041	15717	17344		
19	5275	16372	16279		
20	5308	15584	16245		
21	4800	15668	17479		
22	4882	15192	17861		
23	5099	14722	18166		
24	3195	11779	15697		
25	2917	11634	16579		
26	2928	12190	16072		
27	4585	16185	21227		
28	4987	16632	20391		
29	4776	16566	19615		
30	4989	18628	18631		
31	5455	19339	17152		
32	5289	16441	9198	10937	
33	5134	16078	8815	11596	

Table A.2: Number of passages registered per loop detector 26/06/2021

Number of passages registered 27/06/2021	**Lane**				
Loop detector location	**1**	**2**	**3**	**4**	**5**
1	2683	12167	19393	32	
2	3082	12509	19111	3	
3	2969	12267	19021	1	
4	2600	11130	15214	3613	2492
5	2505	10581	15900	1080	4907
6	2689	10906	15969		
7	2761	10902	15823		
8	2580	10644	15483		
9	2831	11402	16995		
10	2764	11279	16552		
11	2807	11573	16862		
12	3021	11721	16503		
13	3058	11719	16392		
14	3165	11749	16102		
15	3027	11797	16393		
16	2918	11396	16314		
17	2826	11772	16571		
18	3129	11843	16255		
19	2890	11522	16752		
20	2845	11007	16683		
21	2393	10849	18003		
22	2200	10611	18418		
23	2120	10412	18689		
24	1719	8855	14285		
25	1720	8691	14884		
26	1724	9067	14499		
27	3014	12286	18266		
28	3150	12592	17829		
29	3078	12818	16798		
30	3379	14840	15578		
31	3894	15501	14109		
32	3781	13482	7367	8836	
33	3835	13086	7137	9218	

Table A.3: Number of passages registered per loop detector 27/06/2021

Number of passages registered 28/06/2021	**Lane**				
Loop detector location	**1**	**2**	**3**	**4**	**5**
1	5952	18982	22810	41	
2	6606	19613	22378	6	
3	6415	19026	22429	3	
4	5708	16803	17322	4860	4585
5	5317	16136	18265	1793	7441
6	5561	16864	17935		
7	5723	16787	17823		
8	5546	16375	17450		
9	6039	17757	19224		
10	6090	17537	18649		
11	6313	17768	18936		
12	6457	17922	18656		
13	6536	18144	18326		
14	6672	18091	18078		
15	6683	18047	18239		
16	6550	17629	18085		
17	6477	17947	18559		
18	6685	17954	18374		
19	6286	17733	18933		
20	5932	17102	19036		
21	5290	16866	20791		
22	4928	16475	21605		
23	4557	16009	22446		
24	3914	13779	16200		
25	3930	13587	16866		
26	4012	14397	16048		
27	6500	19544	21985		
28	7106	20283	20628		
29	7004	20083	19849		
30	7350	21087	19922		
31	7871	20241	19878		
32	7015	16755	12071	12028	
33	6893	16155	11826	12758	

Table A.4: Number of passages registered per loop detector 28/06/2021

Number of passages registered 29/06/2021	**Lane**				
Loop detector location	**1**	**2**	**3**	**4**	**5**
1	6587	19790	23038	46	
2	7367	20403	22577	2	
3	7030	19781	22645	5	
4	6169	17600	17243	5002	4890
5	5804	16846	18180	1892	7920
6	6162	17391	17991		
7	6146	17415	17915		
8	6028	17167	17371		
9	6582	18449	19303		
10	6634	18181	18792		
11	6937	18481	19004		
12	7038	18650	18721		
13	7175	18685	18547		
14	7199	18775	18269		
15	7175	18789	18388		
16	7103	18301	18216		
17	7047	18712	18565		
18	7361	18555	18454		
19	6697	18431	19176		
20	6333	17685	19428		
21	5555	17592	21147		
22	5291	17281	21818		
23	4953	16775	22692		
24	4222	14381	16062		
25	4187	14006	16923		
26	4268	14920	16052		
27	6950	20191	22458		
28	7547	20976	21057		
29	7444	20640	20342		
30	7873	21660	20429		
31	8230	20484	20843		
32	7319	16844	12634	12629	
33	7046	16233	12456	13438	

Table A.5: Number of passages registered per loop detector 29/06/2021

Number of passages registered 30/06/2021	**Lane**				
Loop detector location	**1**	**2**	**3**	**4**	**5**
1	6947	20895	23617	50	
2	7831	21606	23026	6	
3	7602	20864	23066	7	
4	6599	18582	17644	5237	4962
5	6277	17801	18489	2040	8055
6	6571	18340	18280		
7	6444	18463	18331		
8	6342	18037	17919		
9	7140	19455	19471		
10	7189	19123	18950		
11	7483	19429	19187		
12	7519	19534	19040		
13	7562	19691	18831		
14	7734	19694	18492		
15	7748	19683	18629		
16	7665	19245	18423		
17	7608	19683	18773		
18	7867	19752	18498		
19	7302	19416	19275		
20	6777	18780	19535		
21	6103	18392	21539		
22	5676	17985	22423		
23	5315	17584	23254		
24	4586	15093	16583		
25	4530	14835	17360		
26	4658	15604	16604		
27	7916	21646	21426		
28	8744	21921	20335		
29	8781	21315	19817		
30	8994	22174	20122		
31	9181	21263	20627		
32	7886	17040	12687	13218	
33	8018	16147	12329	14015	

Table A.6: Number of passages registered per loop detector 30/06/2021

Number of passages registered 01/07/2021	**Lane**				
Loop detector location	**1**	**2**	**3**	**4**	**5**
1	7789	21780	24494	50	
2	8678	22507	23808	5	
3	8368	21964	23832	3	
4	7306	19453	18311	5407	5241
5	6995	18743	19136	2143	8308
6	7294	19258	19042		
7	7267	19281	19057		
8	7073	18812	18680		
9	7652	20464	20545		
10	7908	20180	19740		
11	8009	20591	20146		
12	8081	20617	19964		
13	8220	20694	19750		
14	8399	20740	19358		
15	8513	20609	19527		
16	8382	20134	19334		
17	8286	20638	19708		
18	8545	20698	19462		
19	7926	20432	20215		
20	7499	19602	20520		
21	6729	19525	22383		
22	6228	19151	23268		
23	5733	18606	24396		
24	4949	15960	17105		
25	4893	15650	17982		
26	5052	16499	17040		
27	8156	21992	23686		
28	8710	22844	22327		
29	8546	22437	21609		
30	8855	23473	21836		
31	9360	22061	22292		
32	8125	18178	13703	13660	
33	7975	17358	13480	14497	

Table A.7: Number of passages registered per loop detector 01/07/2021

Number of passages registered 02/07/2021	**Lane**				
Loop detector location	**1**	**2**	**3**	**4**	**5**
1	9036	23132	25405	46	
2	10067	23837	24707	4	
3	9739	23408	24583	9	
4	8759	21150	18814	5471	5073
5	8481	20339	19805	2154	8126
6	9079	20927	19436		
7	9131	21079	19191		
8	8969	20397	18840		
9	9650	21974	20901		
10	9706	21750	20128		
11	9982	22217	20408		
12	10248	22101	20217		
13	10158	22169	20202		
14	10453	22201	19682		
15	10399	22232	19879		
16	10283	21738	19598		
17	10251	22125	20124		
18	10479	22103	19936		
19	9914	21881	20649		
20	9398	21191	20851		
21	8522	21158	22807		
22	7964	20850	23752		
23	7514	20497	24554		
24	6755	17987	17841		
25	6816	17768	18618		
26	7002	18622	17622		
27	10033	24165	23847		
28	10748	24670	22663		
29	10611	24121	21945		
30	11124	25281	21934		
31	11570	23899	22552		
32	10481	19636	13604	14169	
33	10214	18963	13301	15034	

Table A.8: Number of passages registered per loop detector 02/07/2021

Number of passages registered 03/07/2021	**Lane**				
Loop detector location	**1**	**2**	**3**	**4**	**5**
1	3570	14803	21927	34	
2	4111	15443	21457	1	
3	4048	14938	21439	2	
4	3503	13478	17020	4360	3065
5	3298	12795	17816	1491	5882
6	3546	13068	17812		
7	3457	13234	17746		
8	3284	12949	17327		
9	3567	13895	19056		
10	3615	13582	18613		
11	3738	13929	18897		
12	3844	14151	18560		
13	3858	14296	18365		
14	3978	14341	18022		
15	4025	14288	18194		
16	3874	13913	18016		
17	3825	14212	18436		
18	4130	14454	17962		
19	3894	14091	18470		
20	3617	13492	18590		
21	3108	13245	20112		
22	2899	13106	20496		
23	2766	12794	20964		
24	2266	10920	16086		
25	2337	10752	16626		
26	2391	11240	16135		
27	4017	15232	20986		
28	4314	15832	20116		
29	4202	15909	19153		
30	4587	17870	18018		
31	5233	18398	16560		
32	4841	15907	8542	10814	
33	4883	15370	8305	11334	

Table A.9: Number of passages registered per loop detector 03/07/2021

Number of passages registered 04/07/2021	**Lane**				
Loop detector location	**1**	**2**	**3**	**4**	**5**
1	2808	12429	19601	29	
2	3163	12834	19272	1	
3	3044	12622	19162	3	
4	2713	11441	15084	3491	2849
5	2580	10885	15728	1124	5148
6	2851	11068	15747		
7	3286	12067	14255		
8	3266	11691	13972		
9	3695	12198	15316		
10	3899	11816	14983		
11	3974	12151	15112		
12	3807	11804	15649		
13	3527	11666	16008		
14	3267	11811	15991		
15	3209	11723	16274		
16	3078	11451	16155		
17	3006	11799	16398		
18	3258	11907	16069		
19	3001	11637	16503		
20	2750	11161	16682		
21	2377	10971	17859		
22	2186	10730	18270		
23	2072	10531	18631		
24	1728	8916	14197		
25	1740	8636	14778		
26	1730	9124	14390		
27	2987	12207	18636		
28	3200	12729	17933		
29	3016	12919	17043		
30	3372	15029	15718		
31	4013	15854	13912		
32	3816	13965	7290	8661	
33	3918	13519	7009	9129	

Table A.10: Number of passages registered per loop detector 04/07/2021

A.4. Final overview data before and after filtering

		Before Filtering											
		Segment											
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Nr of trajectories	**26/06/2021**	12587	12500	12723	12606	13236	11797	12799	11672	13970	12355	13223	12276
	27/06/2021	10492	10438	10603	10527	10985	9862	10546	9705	11457	10081	10741	10021
	28/06/2021	14795	14750	15000	14836	15537	13844	14800	13599	16392	14264	15154	14191
	29/06/2021	15487	15408	15667	15512	16234	14281	15395	14066	17024	14804	15705	14781
	01/07/2021	16712	16620	16916	16730	17494	15404	16544	15211	18233	15697	16671	15636
	02/07/2021	17159	17075	17376	17214	17957	16001	17103	15734	18599	16224	17238	16178
	03/07/2021	12641	12549	12769	12659	13206	11739	12575	11555	13857	12196	12994	12129
	04/07/2021	10815	10753	10917	10822	11297	10100	10777	9892	11789	10455	11149	10392

Table A.11: Number of Flitsmeister trajectories per day, per segment before filtering

		After Filtering											
		Segment											
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Nr of trajectories	**26/06/2021**	4181	4082	4127	4172	4164	4148	4116	4063	4036	888	4111	1549
	27/06/2021	3509	3410	3489	3523	3518	3489	3472	3499	3489	825	3498	1402
	28/06/2021	4793	4730	4760	4797	4792	4799	4659	4735	4660	805	4683	1368
	29/06/2021	4817	4739	4765	4797	4783	4810	4643	4766	4660	796	4658	1318
	01/07/2021	4901	4899	4887	4905	4837	4963	4339	4923	4740	898	4589	1439
	02/07/2021	5936	5828	5844	5927	5902	5943	5570	5855	5733	1070	5701	1697
	03/07/2021	3985	3916	3963	3994	3973	3970	3864	3948	3938	915	3945	1532
	04/07/2021	3350	3276	3335	3359	3359	3336	3293	3312	3331	802	3345	1324

Table A.12: Number of Flitsmeister trajectories per day per segment after filtering

A.5. Formulas for creating bearing of the vehicle

The formula used to calculate the heading (bearing) of the vehicles at each data point is the following, as suggested by Bullock, 2007, in which A and B represent two consecutive data points of the vehicle.

$$\Delta L = longitude_B - longitude_A$$

$$X = cos(latitude_B) * sin(\Delta L)$$

$$Y = cos(latitude_A) * sin(latitude_B) - sin(latitude_A) * cos(latitude_B) * cos(\Delta L)$$

$$bearing = arctan^2(X, Y)$$

After this, the resulting bearing is transformed from radians to degrees. Lastly, since the heading of the last data point of each vehicle in the section is not able to be calculated according to the above mentioned rule since there is no next data point to use in order to calculate the heading, the heading of the final data point is set equal to the average of the headings of the previous two data points.

Next, the heading difference between two consecutive data points is computed according to the headings previously calculated. This is done according to the following formula:

$$\Delta heading_i = heading_{i+1} - heading_i$$

Here again, for the last data point of each trajectory in the road section, the heading difference is taken as the average of the previous two heading differences.

Lastly, the heading difference between the vehicle and the centerline is calculated for every data point of the vehicle. This is done by first calculating the bearing of the centerline in a similar manner as the heading of the vehicles was calculated, namely through the formula by Bullock, 2007. Then the difference in heading between the vehicle and the centerline at the nearest point is calculated for every data point of each vehicle.

A.6. Formula for Balanced Accuracy:

A.6.1. Binary Classification

Balanced accuracy is a way to calculate the accuracy from a confusion matrix in binary classification, when the size of the two categories is largely different. It is calculated in the following way:

$$Balanced\_accuracy = (Sensitivity + Specificity)/2$$

In which:

$$Sensitivity = True\_positives/(True\_positives + False\_negatives)$$

and

$$Specificity = True\_negatives/(True\_negatives + False\_positives)$$

A.6.2. Multiclass Classification

In multiclass classification the balanced accuracy is calculated as follows for a case with four classes:

$$Balancedaccuracy = (Recall\_class\_1 + Recall\_class\_2 + Recall\_class\_3 + Recall\_class\_4)/4$$

In which recall is calculated the same as sensitivity for each class.

A.7. Extraction of rule-based method from Random Forest Model

In order to evaluate the extent to which a rule can be deduced from a random forest model, an experiment is conducted using the definition for lane change recognition as created by Van Ballegooijen (2019). This rule, using the delta heading of a vehicle for a number of consecutive time steps is the following:

$$(\Delta h)\ at\ t_0,\ t_1,\ and\ t_2 < 0\ or\ (\Delta h)\ at\ t_0,\ t_1,\ and\ t_2 > 0$$

$$and$$

$$abs(\Delta h_{t_0} + \Delta h_{t_1} + \Delta h_{t_2} + \Delta h_{t_3}) >= 6$$

By labelling a data set of Floating Car Data according to this rule, it is then possible to train the random forest classification model on this labelled data. Due to the researchers knowledge of the rule used in order to label the data, it can be examined whether this rule has been found by the random forest model.

In order to do so, a set of 11 values, of which two are variables (a and d), which are altered, is manually input (e.g. [0,0,0,a,0,d,0,0,0,0,0]). The values of the fixed numbers is either 0 or 1, and the location of the variables within the range can alter. By changing the value of the two variables, ranges of values are created, which sometimes do and sometimes do not fulfill the above shown lane change recognition rule. For each range it is then found whether or not the model indicates a lane change. Figures A.1, A.2, A.3 below depict a few examples of some ranges tested, in which a blue dot indicates the model recognises a lane change, and a red dot indicates no lane change is recognised.

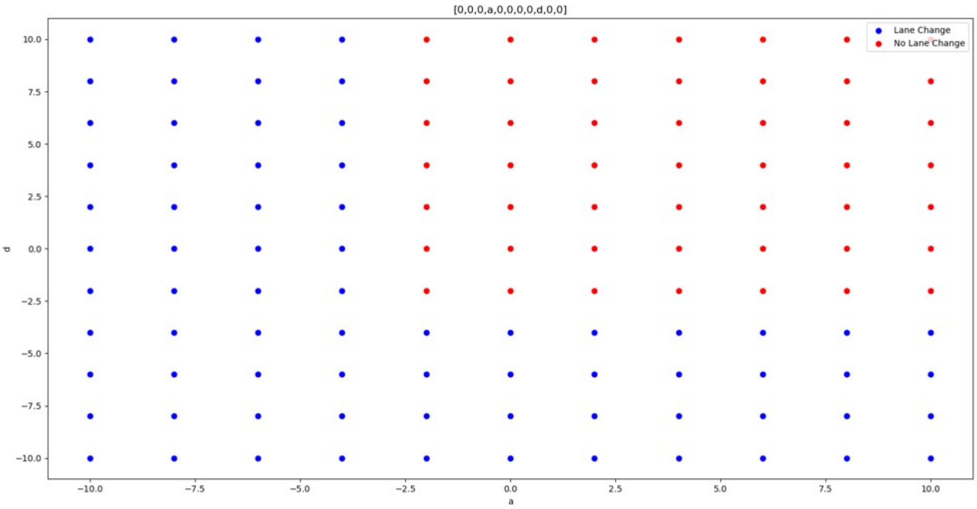


Figure A.1: Random Forest rule recognition 1

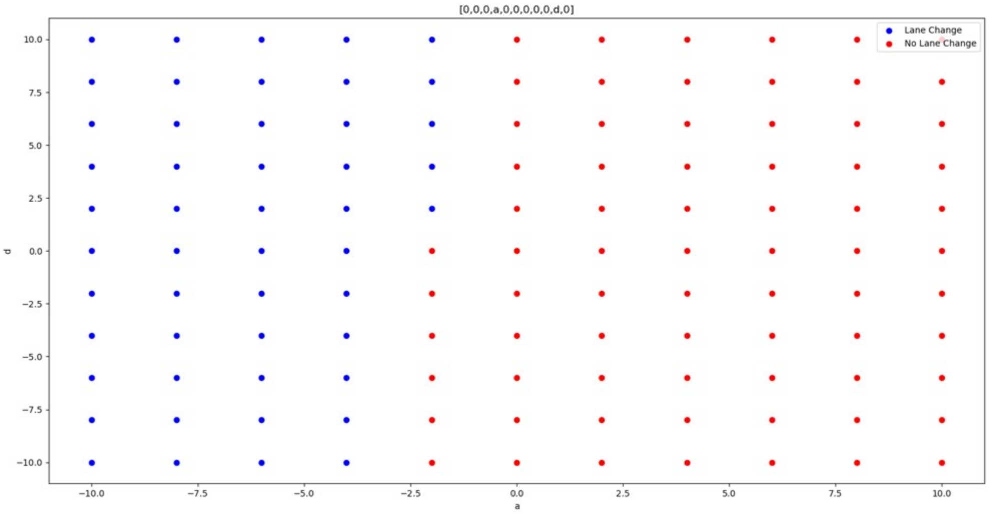


Figure A.2: Random Forest rule recognition 2

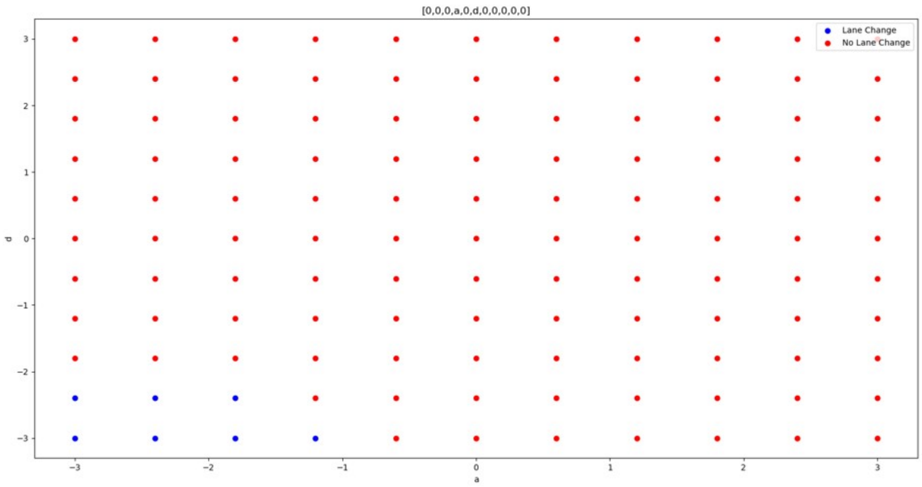


Figure A.3: Random Forest rule recognition 3

From the analysis of when the model recognises lane changes or not, it is found that the rule as implemented for the labels, is not recognised by the random forest.