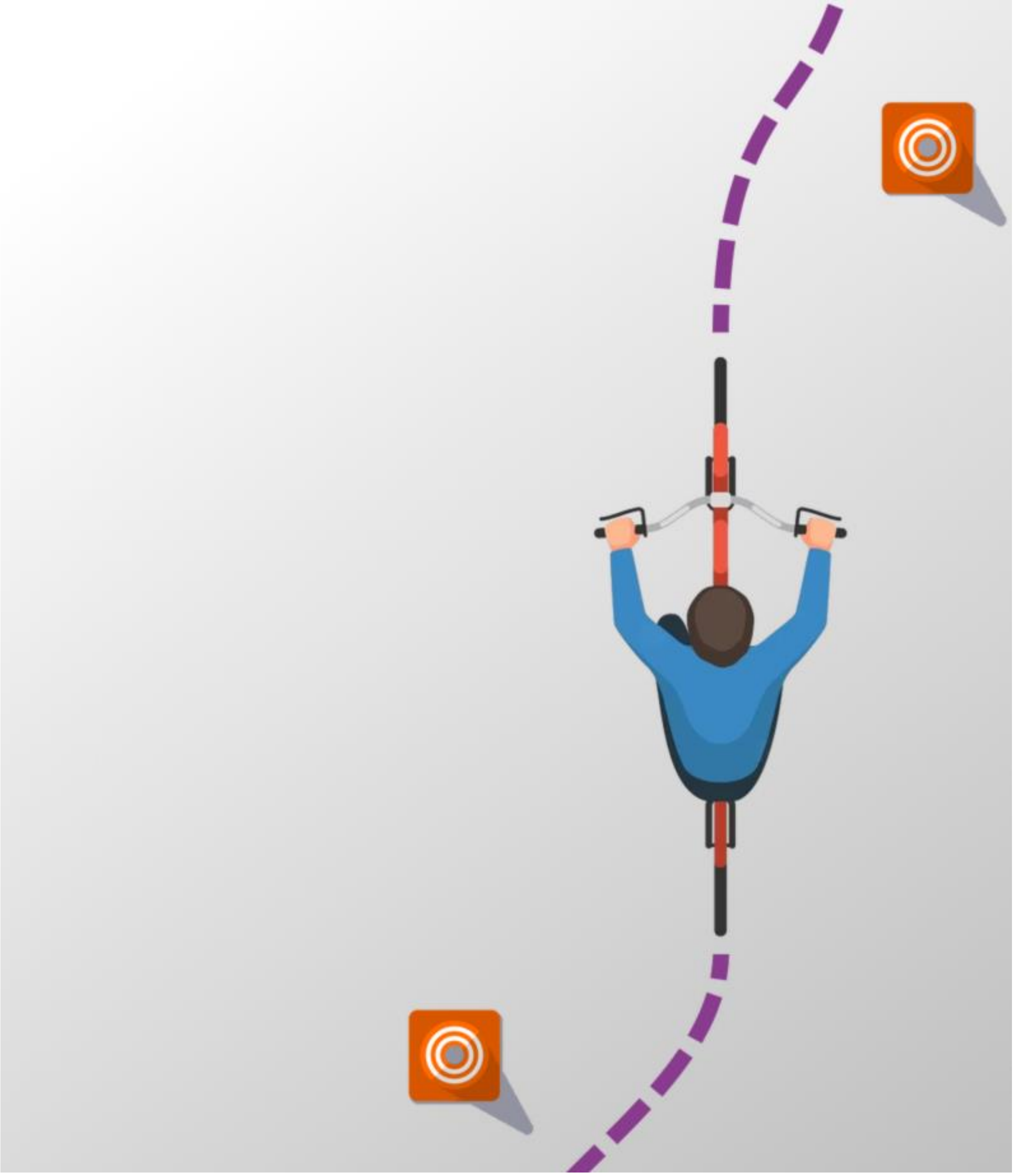
Cycling Maneuver Prediction Using Bicycle-mounted Motion Sensors

BSc thesis by Gijs de Smit



University of Twente
Advanced Technology
Pervasive Systems
22-06-2023

Committee members:
Deepak Yeleshetty
Paul Havinga
Frank Wouda

ABSTRACT - There has been a growing number of road accident fatalities involving cyclists. Timely prediction of a cyclist's maneuver can be useful to proactively avoid potential accidents. This research focuses on utilizing bicycle-mounted motion sensors for maneuver prediction. The goal is to find out to what extent the maneuvers of left turn, right turn and cruise can be predicted using a Convolutional Long Short-Term Memory Neural Network (CNN-LSTM). A literature survey and preliminary experiment are conducted to motivate sensor placement on the handlebar, which is compared with placement on the frame. An experimental setup is designed where 20 participants performed the three cycling maneuvers on an artificial intersection outlined with cones. The data collection method consists of a custom Android app to record IMU data of a smartphone and a Thingy:52 sensor. The model input is optimized by removing unnecessary features, adding processed features and low-pass filtering. The final model was able to predict the cycling maneuvers 0.25 s in advance using a 1 second time window with an F1-score of 0.92. Lastly, future research can be done into combining various sensors and placements to increase the prediction time gap and improve classification performance.

1 INTRODUCTION

Travelling by bicycle is a common mode of transport in the Netherlands, accounting for 25% of all trips made there [1]. The Netherlands witnessed a significant 40% increase in the amount of yearly road accident fatalities involving cyclists in 2022 [2]. Consequently, it is imperative to improve cycling safety in traffic, for example by introducing new technologies that help prevent accidents.

Passive protection such as helmets or protective braces may reduce the severity of cycling accidents, but this type of equipment is reactive and by definition cannot proactively prevent accidents. However, it might be possible to reduce cycling accidents by using active protection systems which are not yet adapted in traffic on bicycles. An active protection system continuously monitors the ongoing movement or surroundings of the bicycle and can intervene when necessary. Such systems, could rely on active cycling behavior monitoring or maneuver prediction, for which several methods already exist. For example, recognising dangerous cycling behavior [3] and predicting cycling maneuvers based on head motion [4], where the better approach to prevent an accident is maneuver prediction.

Current research shows it is possible to predict a cyclist's maneuver [4]. The existing method to predict cycling maneuvers depends on placing motion sensors on the cyclist's helmet. Consequently, the system relies on the user making an effort to wear the sensors. Nevertheless, a cyclist might not always choose to wear the sensors. Body-worn sensors could pose an inconvenience and might also hinder cycling performance. This paper envisions a less invasive approach without user-dependency that could yield a similar result. Mounting the sensors on the bike removes the dependency of user-interaction with the system and cycling behavior could potentially not be affected. Furthermore, bicycle-mounted motion sensors have been proven to have many use cases as will be mentioned in section 2. It might be useful to exploit them further. Using such motion sensors also opens up various opportunities, such as fall detection [5] and road surface condition mapping [6].

In short, bicycles equipped with motion sensors could be used for maneuver prediction, which in turn could contribute to safer cycling. This paper aims to add to the field of research of bicycle maneuver prediction, by considering bicycle-mounted motion sensors. The following subsection will elaborate the goal and scope of this work.

1.1 Goal & Scope

The goal of this paper is to investigate how IMU (Inertial Measurement Unit) sensors can be used to predict maneuvers by placing them on the bicycle. Firstly, a literature survey is conducted to find a good sensor placement, followed by a preliminary experiment to identify features as pre-maneuver indicators in the motion data using this placement. This research is limited to the three maneuvers of turn left, turn right and cruise.

For the data analysis part, an existing CNN-LSTM model is used to predict the different cycling maneuvers. The focus in the experimental method lies on gathering enough data from participants to train this model, which exploits features in bicycle motion data. The performance of the model will be evaluated with metrics such as F1-score and prediction time among others. To make a contribution to the existing model, investigations will be conducted on refining the input data and architecture of the existing model. Besides that, two distinct sensor placements are compared. Furthermore, this work will not optimize computation time or memory usage of the classification model, because it lies beyond the scope this paper.

The rest of the paper is structured as follows. In the next subsection the research questions are stated. Section 2 discusses similar studies by presenting an overview of the two main categories of research. Following that, section 3 will cover the methodology used to perform this research project and section 4 describes the experiment. Finally, results are presented in 5 and will be discussed in 6 followed by the conclusions in 7.

1.2 Research Questions

To address the problem a main research question is formulated along with three sub-questions. The main research question is: *To what extent can cycling maneuvers be predicted from bicycle-mounted IMUs using deep learning?* The problem is split into the following sub-questions:

1. Where to place IMUs on the bicycle and how does this placement capture features that indicate maneuver intention?
2. How to select and process input data for a CNN-LSTM model that can classify cycling maneuver intention?
3. To what extent and how early can the model predict cycling maneuvers based on F1-score?

The sub-questions 1, 2 and 3 will be answered in sections 3.1, 5.2.3 and 5.2.1 respectively.

2 RELATED WORK

Gaining insights from cycling using motion sensors has been a growing field of research. The choice of motion sensor is often an IMU in combination with a smartphone. IMUs are low-cost, effective and easy to process and smartphones are equipped with this sensor. In addition, the smartphone can be used to store and process recorded data while cycling. Previous research concerning cycling motion inference can roughly be classified into two categories: *behaviour & characteristic monitoring* and *maneuver prediction*. The following two subsections will cover these categories and highlight parts that helped shaping the method.

2.1 Behaviour & Characteristic Monitoring

Cycling behaviour is defined as the way a cyclist moves themselves or the bicycle they are riding. There have been studies investigating different kinds of cycling behaviours with the objective to improve cycling safety by detecting dangerous riding behaviour [3, 7]. In [7] the 6 dimensions of an IMU of a handlebar-mounted smartphone were used to classify the four bicycle behaviours of stop, run straight, turn right, and turn left. Using a random forest classifier they obtained an F-measure of 0.9. Their main focus was on identifying the types of noise from road surface, handlebar and drift and mitigating these using a low-pass filter among other things. These papers show that a smartphone on the handlebar can be used for the method of this work and a low-pass filter might influence classification performance of the model.

BikeMate is a bicycling behavior monitoring system proposed by Gu et al. [3]. Similar to [7], BikeMate uses the IMU of a smartphone, but also the GPS to the infer dangerous riding behaviors of lane weaving, standing pedaling and wrong-way riding. A support vector machine model was implemented with an accuracy of roughly 90%. BikeMate inspired the idea of exploring GPS speed data as input for the model used in this paper.

A less obvious insight obtainable from bicycle mounted motion sensors is a cyclist's weight as proposed by Benitah in [8]. Benitah compared several machine learning models and found that logistic regression is best to classify three different weight classes. In addition, it is discussed that accuracy could be improved by gathering more data. The data collection method of using a custom Android app is based on this paper.

DoubleCheck [9] designed a non-intrusive and reliable detection method for single-hand cycling. The method consists of using the IMU of a handlebar-mounted smartphone to detect the six combinations between the three handgrip situations of left, right and both hands and two different road surfaces. DoubleCheck gives rise to the idea that using a hand to indicate turn direction can be obtained using a smartphone on the handlebar, besides further addressing the effect road noise.

2.2 Maneuver Prediction

A somewhat newer and less explored area of research is predicting cycling maneuvers. The difference of detecting maneuvers compared to maneuver prediction is that the latter focuses on predicting certain cycling maneuvers before they occur rather than during. HeadMon [4] utilizes the head dynamics of a rider to predict the four cycling maneuvers of: cruise, left turn, right turn and right lane change. This study proves that gaze is an indicator of maneuver intention in traffic. To track a cyclist's gaze they made use of head dynamics, which they measured using a helmet-mounted IMU. To store the head motion data a smartphone was used mounted on the handlebar and the ground truth was manually labelled using recorded video. The classification model was an attention-based CNN-LSTM, which achieved an accuracy of 85%, 4 seconds before the occurrence of a maneuver. However, as mentioned in section 1 this is an invasive way of sensing, which could be inconvenient for the cyclist. Also, Gaze can be an unreliable indicator as a cyclist can still look around and not turn, which is a false positive. From HeadMon it is gathered that a CNN-LSTM model is a fitting choice for the methodology of this work. The maneuvers in this paper will be predicted using a similar model architecture.

3 METHODOLOGY

In this section the methods that have been used to conduct this research project are described. Specifically, this section will provide an answer to sub-question 1 and describes the methods used to address the other two sub-questions.

3.1 Sensor placement

The following two subsections will pose an answer to the sub-question 1: "Where to place IMUs on the bicycle and how does this placement capture features that indicate maneuver intention?", which is motivated by current literature and a small experiment.

3.1.1 Literature. As discussed in '2 Related Work', existing studies have used IMUs on the bicycle for different applications. This paragraph will summarise and compare relevant sensor placements from related work. [7] motivated motion sensor placement on the handlebar instead of rear wheel axle, where the most noticeable motions regarding simple behaviours like turning and cycling straight occur. For this reason, it is suspected that this placement can capture potential pre-maneuver behaviours such as slowing down or turning the bicycle. In addition, DoubleCheck [9] proved that a handlebar-mounted smartphone can be used to detect single-hand cycling based on handgrip. Therefore, a sensor on the handlebar can detect when a cyclist is using an arm to indicate turn intention. On the contrary, a study investigating the weight of a cyclist exploited placement on the back of the bike under the cyclist, as that is where the weight has the most impact on the measured acceleration [8]. This placement is not suitable for this research and will not be considered.

Before entering a turn, a bicycle rider consciously or subconsciously performs a 'countersteer' to maintain balance [10]. This counterintuitive maneuver arises from the fact that the center of mass of the rider and vehicle must lie inside the curve of the turn. To perform a countersteer, the rider must adjust the handlebars in the opposite direction of the intended turn. For that reason, a handlebar-mounted sensor would be able to capture this behaviour.

In summary, a handlebar-mounted motion sensor is used to detect behaviours like turning and might also capture maneuver indicators like countersteering and indicating with an arm. Consequently, this work will verify whether an IMU on the handlebar can capture features regarding maneuver intention using a preliminary experiment.

3.1.2 Preliminary Experiment. A small experiment was conducted to investigate whether an IMU placed on the handlebar captures

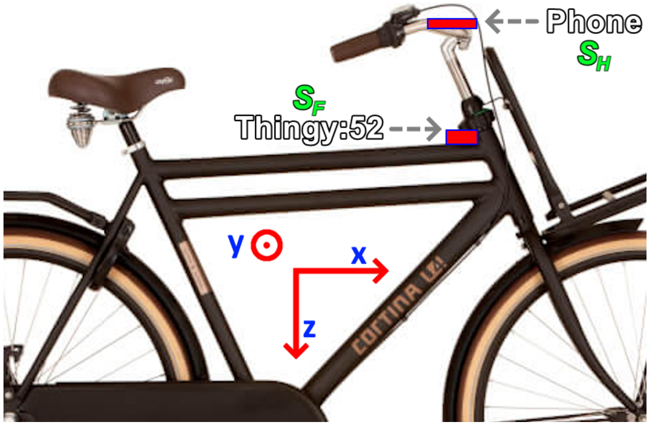


Fig. 1. Bicycle Configuration with S_H and S_F with coordinate system

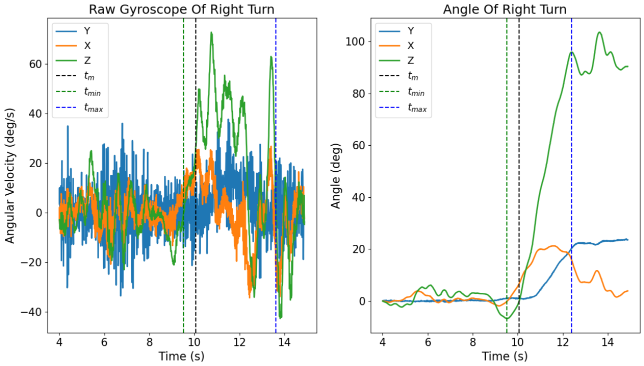


Fig. 2. Gyroscope of Phone mounted on handlebar of a right turn

features that showcase cycling intention. Using the equipment and data collection method explained in sections 4.1 and 4.3, motion data was gathered of the three cycling maneuvers of right turn, left turn and cruise. A cycling road on the University of Twente campus was used that has an intersection with a sharp turn. This turn was approached from both sides for left and right turn maneuvers, where each maneuver was measured several times. Acceleration, angular velocity and angle data were compared to identify consistent pre-maneuver indicators.

Figure 1 shows the bicycle and coordinate system used. More details on this figure will be given in section 4.1. For this experiment only the smartphone on the handlebar, denoted S_H was used. The X angle represents the tilt of the bike, the Y angle indicates whether the bike is being driven uphill or downhill and the Z angle describes the steering angle. The angle is obtained by integrating angular velocity of the gyroscope. Integration is started at the mean of angle Z in such a way that this angle oscillates around 0 degrees, where a Z angle of 0 means the handlebar is straight.

Figure 2 shows the motion of a single right turn. The left plot is the raw gyroscope data and the right plot is the angle upon integrating the raw gyroscope. The start time t_m of a turn maneuver is defined as the time when the angle of the Z dimension last crosses zero in the graph plotted over time. Meaning, the vertical dotted line indicates when the cyclist starts turning right, and doesn't stop doing so until the end of the turn. This is the last time the angle is aligned with the approach of the bicycle to the turn. Besides t_m, two peaks in the Z angle before and after the turn are also extracted, denoted t_{max} and t_{min}. Note that order of t_{max} and t_{min} in time depends on the turn direction.

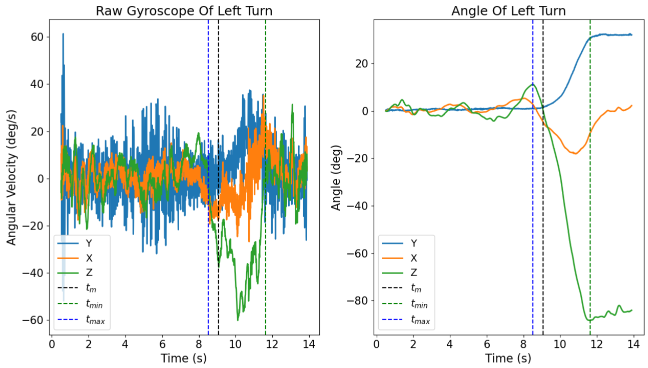


Fig. 3. Gyroscope of Phone mounted on handlebar of a left turn

Furthermore, right before t_m, a countersteer can be seen in the Z angle, showing that the cyclist steers left before entering the right turn. Part of the countersteer is that the X angle peaks in the same direction as the Z angle. This confirms that a countersteer is used to initiate leaning (tilting) in the direction of the turn. Additionally, Figure 3 shows a left turn and confirms that the gyroscope's X and Z dimensions are pre-maneuver indicators. When making a left turn, the peaks in the X and Z angles are mirrored compared to a right turn.

Figure 4 shows the motion of a cruising maneuver. Note that the axes are scaled differently compared to the previous figures. Looking at both the angle and angular velocity plots, somewhat stable oscillating patterns of similar frequency can be seen in all three dimensions. This pattern is suspected to be caused by the cyclist's need to correct for the imbalance caused by pedaling while riding straight. In the figures for the left and right turn maneuvers, this oscillation appears before and after the turn. The amplitude of this oscillation was found to decrease if the cyclist stops pedalling.

Adding to that, drift noise is present in the angle over time. This is caused by the error when integrating the angular velocity. In several stationary measurements, this noise was investigated and determined to be negligibly small. Roughly 1.2 degrees offset in 60 seconds. Potentially a high-pass filter can be used to remove drift noise in the angle. Nevertheless, a HPF is not applied, since it removes important low-frequency features.

The Y dimension of the angular velocity and angle doesn't show any different features for both turns. However, the acceleration in the Y dimension might be relevant, since this shows how the bike slows down or speeds up in the direction of travel.

From the three dimensions of acceleration no features could be qualitatively determined. However, it is suspected that accelerometer data still might have features. Therefore, for the machine learning

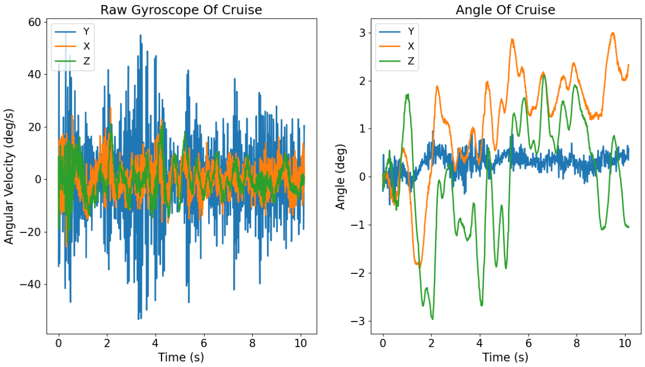


Fig. 4. Gyroscope of Phone mounted on handlebar of cruising

model, this data is fed as input and input features are selected based on classification performance.

In conclusion, an IMU placed on the handlebar captures features that indicate maneuver intention. Specifically, two dimensions of the gyroscope exhibit peaks before a turn occurs. As for the accelerometer, further investigation will be performed in the machine learning model section. For comparison, a second sensor is added on the frame of the bicycle that will be referred to as S_F.

3.2 Machine learning classification model

This work considers predicting maneuvers as a multi-class classification problem. In the literature discussed in section 2, machine learning classification models have been proven to be a strong approach to solve such problems. HeadMon [4] used a deep learning CNN-LSTM model for maneuver prediction and such a a model seems most fit from related work. The input for their model is a time window of IMU data of a head-mounted motion sensor and the output is the maneuver that is about to happen. This work does something similar and therefore will base its model on the one proposed by HeadMon. There are potential improvements to the model that this paper will investigate. HeadMon doesn't look into noise filtering and just uses the raw accelerometer and gyroscope data without explanation. Thus, this study will evaluate the effect a low-pass filter has on model performance. In addition, existing features can be removed or new processed features can be added to the model.

Figure 5 shows the architecture of the CNN-LSTM model that will be used for maneuver prediction in this research, based on the one proposed by HeadMon. The model is programmed in Python using the deep learning framework Keras and the NumPy package is used for processing data. The figure shows that the input layer receives a time window of bicycle movement data. The shape of an input sample is the number of data points in the window W by the number of features F. Each feature is normalised by removing the mean and scaling to unit variance. The feature extraction network contains a pair of Conv1D layers, each followed by Relu activation and a MaxPooling1D layer. The purpose of the convolutional layer is to extract low-level features from the data. The first and second convolutional layers have output size 16 and 4. Both layers have kernel size 7 and padding mode of 'same'. Maxpooling is used to extract essential features by downsampling, using pool size is 2. The convolutional part is responsible for finding spatial dependencies, whereas the attention and LSTM layers are is responsible for finding temporal dependencies. This part starts with an LSTM layer followed by a tanh activation and a dropout layer to prevent overfitting. The attention layer after chooses which features are highly correlated for final classification. Lastly, a Softmax classifier outputs one of the classes. Cross-entropy error is used as the loss function and the Adam algorithm is used to optimize the model.

The structure and parameters of this model will be tuned to investigate whether a smaller model can reach significant results. Deep learning models typically require more data for classification compared to simpler models like Support Vector Machines. Therefore, it is important that enough training samples are collected. This will be explained further in the experiment section.

The classes that the machine learning model will attempt to detect are that of left turn, right turn and cruise maneuvers as shown in Table 2. These are chosen since they are the essential maneuvers performed by cyclists while commuting and they indicate bicycle trajectory. For an active cyclist protection system to work, it's key that the model is be able to make a distinction between cruising and turning.

3.3 Evaluation metrics

To know how the machine learning models perform, several metrics are evaluated. Firstly, the data is split up into a training and a testing set to prevent overfitting and performance will be evaluated on the the testing set.

Secondly, there are several time parameters that play an important role in this work. From the motion data of a turn maneuver, a single time window can be extracted around when the maneuver occurs. Figure 6 shows how a prediction window is defined. G defines the time gap between the window end and maneuver start and W defines the window size, where both are expressed in seconds. How soon and how long a cyclist shows motion features before a maneuver is not known. Therefore, the performance of the model will be evaluated for the variables of G and W. The end of the detection window t_w can be calculated from $t_m - G$ and the start of the window is obtained from $t_w - W$. Note that if G = 0, it means that there is no time gap and the window consists only of points before t_m. Besides that, evaluating model performance for different combinations of G and W will pose an answer to sub-question 3.

Additionally, F1-score for each class will be assessed for the above mentioned time parameters. F1-score is calculated from precision and recall. Precision is the proportion of correctly predicted samples (true positives) of a class out of all samples predicted as that class, see equation 1. The false positive count is defined as the number of samples that were incorrectly predicted as positive.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{1}$$

Recall is a measure of the proportion of correctly predicted samples of a class out of all samples belonging to a class. It is calculated as follows:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2}$$

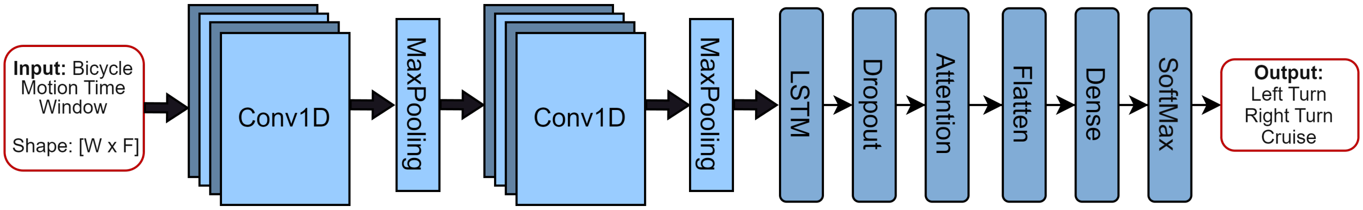


Fig. 5. CNN-LSTM Model Architecture

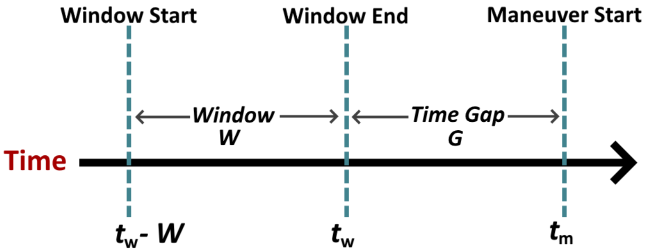


Fig. 6. Time parameters of a prediction window

F1-score captures the trade-off between precision and recall, and is calculated using equation 3.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

The overall F1-score on the testing set is computed as the weighted average of the F1-scores of each class. Besides F1-score, a confusion matrix will be made to help visualise how well the model predicts the individual classes.

Moreover, different combinations of input features will be tested to identify relevant features. Specifically the x, y and z dimensions of the acceleration and angular velocity can be added or removed. Also, the angle obtained by integrating the angular velocity over time can be used an input feature. Similarly, the speed recorded by the GPS can be tested as an input feature.

Finally, it is also investigated what effect a low-pass filter (LPF) has on the model performance for various cut-off frequencies, since removing road noise can improve detecting cycling behaviour [7]. The LPF will not be combined with a high-pass filter, since the frequencies of cycling maneuvers are in the lower range.

4 EXPERIMENT

This section covers the experiment that was performed to gather motion data of participants while cycling. First the necessary equipment is covered, followed by an overview of the experimental setup. Then the method of data collection and processing is elaborated. Lastly, the participant briefing section provides instructions for replicating this experiment.

4.1 Equipment

This section covers the tools, and their specifications, that were used in order to conduct the experiment. To collect and store data a Samsung A70 smartphone was chosen. This phone is capable of gathering both gyroscope and accelerometer data at a sampling frequency of 200 Hz.

A Nordic Thingy:52 sensor is connected over Bluetooth Low Energy to the smartphone. This sensor has a lot of different sensing capabilities, but this research only utilises its gyroscope and accelerometer sensors. The sampling frequency was set to match the phone's sampling frequency of 200 Hz. The Thingy is suited for this experiment because of its compact size and long battery life. Moreover, Nordic Semiconductors provides an Android Library with which applications can be created to fully customise the capabilities of the Thingy:52.

An existing android application from [8] was modified for the objectives of this research. The app collects and stores data of both the phone's IMU and Thingy:52 sensor alongside GPS location and speed. This custom app is chosen since it's important that the cyclist is able to see their speed and conveniently start and stop the measurement. Adding to that, the filename for each measurement can easily be changed. Using this app allows for quick experimenting and reduces the time needed to manually process the data.

The bicycle that was utilized for this experiment is the Cortina U4 Transport (Men's version). A picture of the bicycle configuration can be seen in Fig. 1. The phone is attached to the handlebar using a phone holder and the Thingy:52 is positioned on the front of the frame. This arrangement was selected to enable the comparison between handlebar and frame movement, taking into account the higher amount of noise found in the handlebar. In the following sections, the phone and Thingy:52 will be referred to using S_H (sensor handlebar) and S_F (sensor frame) respectively.

The axes of the bike and sensors can be understood in terms of acceleration as follows. X shows acceleration in the direction of travel (longitudinal), where braking would be most noticeable. Y indicates how the cyclist leans left or right (lateral). Z shows the downwards force (vertical). Similarly, rotation around the axes are understood as follows. Rotation around X is the tilt of the bicycle (roll), rotation around Y is driving up or downhill (pitch) and rotation around Z showcases steering left or right (yaw).

4.2 Setup

Figure 7 shows a sketch of the experimental setup. Yellow and orange cones were placed to form an intersection with a road width of 2 meters. The orange cones indicate the halfway points of the curves and they serve as a physical obstacle for the cyclist to avoid. In addition, blue chalk is used to outline the road width to better guide the cyclist.

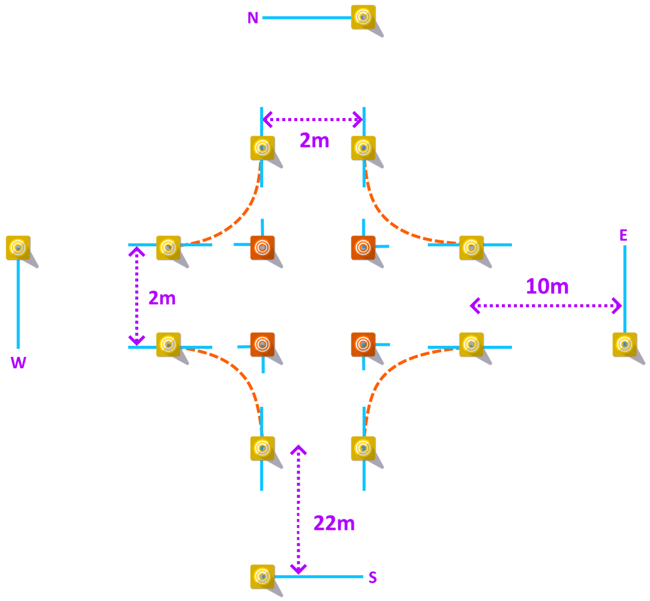


Fig. 7. Experimental Setup: Constructed Intersection (not to scale)

The location of the setup was on the University of Twente campus at Hogekampplein[1]. This location is chosen for its smooth asphalt road surface to reduce road noise, big open area and little amount of traffic. The intersection has two long approaches from the north (N) and south (S) side. For this reason, the participants approached the intersection from N and S to have enough time before the occurrence of a maneuver. Consequently, also preventing overfitting of specific turn direction. The N and S cone are 22 meters spaced from the intersection. Because of space limitation, the west (W) and east (E) cone are spaced 10 meters from the intersection.

Moreover, the setup in Figure 7 is currently configured with sharp 90 degree angle turns. However, the orange cones indicating the turn point can be moved to the orange dotted lines for a wide turn. These turn types are chosen to represent different traffic scenarios.

It should be mentioned that the location has a slight increasing slope from the west to east side. However, this is not a cause for overfitting, since the measurements are performed in such a way to have as many downhill as uphill turns in both directions.

4.3 Data Collection

Data is collected from both S_H and S_F and can roughly be split into three categories: Thingy IMU, phone IMU and phone GPS. Table 1 includes all of the variables that are written to .csv files using the android app. All IMU related data is written with a sampling frequency 200 Hz. The GPS data is written whenever there's a location update, which is roughly 1 Hz while cycling. For each of the three categories there's a designated .csv file. Section 4.4 will cover what is meant by calibrated acceleration.

Table 1. Devices and variables that they record

Device	Variables
Thingy:52 (S_F)	Acceleration x, y, and z (g), Calibrated acceleration x, y, and z (g), Angular velocity x, y, and z (deg/s)
Phone (S_H)	Acceleration x, y, and z (m/s^2), Calibrated acceleration x, y, and z (m/s^2), Angular velocity x, y, and z (rad/s), GPS location (latitude, longitude), GPS speed (km/h)

Table 2 shows the measurements performed by each participant. For a turning maneuver, they approach the start line with a speed of about 10, 15 or 20 km/h from both the north and south side, thus each filename is used twice with a different timestamp. They perform this for both a sharp and wide turn type. The cruising maneuver is repeated four times for each speed category, equally divided over both approaching sides. For a cruise, the participant is told to cycle straight over the intersection with the specified constant speed. All of this results in a total of 36 measurements per participants, split into 12 measurements per maneuver. From each turn maneuver a single time window can be extracted to be fed into the model. How this window is extracted will be explained in section 4.4.1.

Table 2. Classes and Measurements Performed by Participants

Maneuver/class	Approaching Speed (km/h)	Turn Type
Left turn	10, 15, 20	Sharp, wide
Right turn	10, 15, 20	Sharp, wide
Cruise	10, 15, 20	N/A

4.4 Data Processing

The sensors have to be calibrated after connecting the Thingy:52 to the phone over Bluetooth. This means that the bicycle must remain still in an upright position for 5 seconds. In this time, an average of the acceleration of both the phone and Thingy accelerometer in all three dimensions will be collected. After that, calibrated acceleration is obtained by subtracting the average. The app stores both the calibrated and raw accelerometer data.

The Thingy:52 x, y and z axis are exactly like those in Figure 1, which are not directly in line with those of the phone. Therefore, the x, y and z axis of the phone are translated such that both sensors' axes are matched. This operation is not performed by the app, instead this is done while post-processing the data.

4.4.1 Ground-truth labelling and window extraction. A time window is extracted as follows. In a single measurement the cyclist performs a specific maneuver and one window is extracted. As covered in section 3.1.2, t_m is defined as the turn maneuver start time which is when the Z angle of the handlebar last crosses the zero in the graph over time. Section 3.3 further elaborates on the gap size G and window size W of the window based on t_m. G and W are fixed for all windows when they are fed into machine learning model, which enables evaluation of the model in terms of G and W. Moreover, the time window for a cruise maneuver only depends on W and

[1] Hogekampplein Google Maps Location

the end time of a cruise window is fixed for all evaluations. The end time for the cruise windows is set at half the duration of their respective measurement. A window is labelled based on the filename that specifies a right, left or cruise maneuver.

4.5 Participant Briefing

This section details the instructions that are given to the participants in order to replicate this experiment. A participant is first asked for his/her written consent to participate, see section 4.6 for more about the ethics of this experiment. After that, the participant is instructed test drive the bicycle to see whether the saddle needs adjusting for comfort. This also helps the participant get a feeling for the bicycle. Then the participants are told how to start and stop recording data using the app. They're informed to adjust their speed to the specified speed category for a certain measurement by looking at the app interface. The speed category is a guideline, after passing the north or south cone cyclists should not check their speed on the phone. They must take a turn as they would normally, slowing down and turning as they think is fit. A measurement involving a left/right turn is started at the north or south edge of Hogekampplein when the bicycle is facing the intersection with a straight approach, this is roughly 50 meters from the intersection. The cyclist can already be up to speed when pressing start by using a small run up on the big open square. The turn measurement is stopped when the cyclist has passed either the west or east cone, depending on the turn direction. A cruise measurement is performed by cycling straight at a constant speed over the intersection from the south to north cone or vice versa.

Moreover, there a several key aspects to keep in mind. Firstly, the measurements involving the wide turn are performed before the sharp turn measurements. This prevents the cyclist from getting used to the sharper turn and replicating their behaviour for the wide turns. In addition, participants are not told about any cycling motion behaviours that are investigated in this research before the experiment, since this might effect their cycling behaviour.

Finally, the researcher is present at all times so that he can monitor each of the maneuvers and to make sure the participant performs the measurements as instructed. The total of 36 measurements performed by each participant took on average about 30-40 minutes.

4.6 Ethics

This research requires human participants. Therefore, ethical approval was requested from the Ethics Committee Computer & Information Science of the University of Twente[2]. The guidelines outlined by this committee are followed. Written consent is obtained from participants in this research and their motion data is kept anonymous. Participants are fully briefed about the risks and no personal information is gathered. In addition, a requirement for participating is that you should have enough cycling experience to comfortably perform the maneuvers.

5 RESULTS

This section will show the results of the experiment that was performed. First the obtained data set is shown in terms of the amount of collected samples per maneuver. Following that is the model evaluation section, which shows all relevant results obtained from the CNN-LSTM model.

Table 3. Data set, total samples obtained for each maneuver

	All Data	Testing Data (id: 0-5)	Training Data (id: 6-19)
Left Turn	246	75	171
Right Turn	241	72	169
Cruise	239	71	168
Total	726	218	508

Table 4. Hyper-parameters used for model evaluation

Parameter	Value
Batch Size	60
(Initial) Learning Rate	0.001
Numpy Randomizer Seed	42
Time Gap	0.25 s
Window Size	1.0 s

5.1 Data set

In total twenty participants performed the experiment. The goal was to obtain 36 measurements from each participant divided over 3 maneuvers. However, this was not exactly what was obtained. Table 3 shows the number of measurements conducted for each maneuver. In addition, the chosen train and test split is shown. Participants with id from 0 up to and including 5 are used for testing, and 6 up to and including 19 are used for training. The reason why there is no equal division across the three maneuvers is because sometimes a participant took a wrong turn. Instead of discarding such measurements, they were simply relabelled to the correct turn direction. Besides that, t_m was successfully extracted from all turning maneuvers as well as the peak in angle Z at the end of the maneuver. Using these, it was found that the average turn duration in this experiment is 2.15 seconds.

5.2 Model Evaluation

This section shows the evaluation results of the CNN-LSTM model and will help answer research questions 2 and 3. The performance of the model is measured using the testing set F1-score for all evaluations. Table 4 shows the values of the parameters such as batch size and initial learning rate that are kept constant, excluding time gap and window size for the first evaluation. In addition, the training and testing samples are shuffled using a NumPy randomize function with constant seed. The first subsection covers the best choice for the time parameters and fixes them for subsequent evaluations. The second subsection investigates the effect of a low-pass filter on both sensors and compares the two placements. Following that, a selection of raw and processed features is made for which the model has the best performance. The evaluation section is concluded by looking into simplifying the convolutional layers of the model.

[2] CIS Ethics Committee

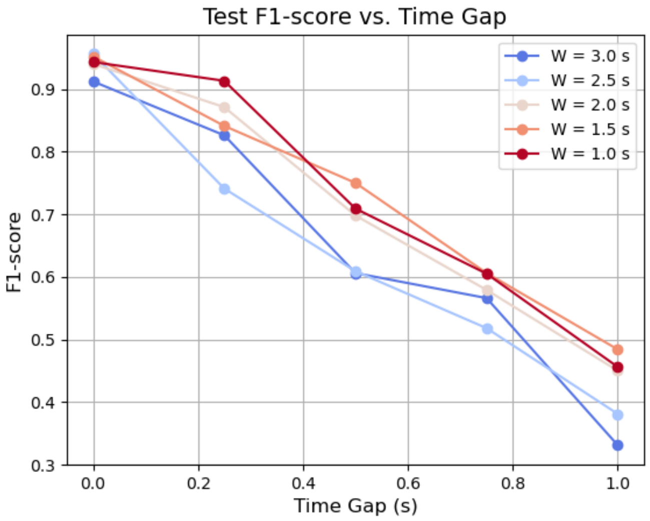


Fig. 8. Testing set F1-score versus time gap for different window sizes, using default model and raw IMU data of S_H

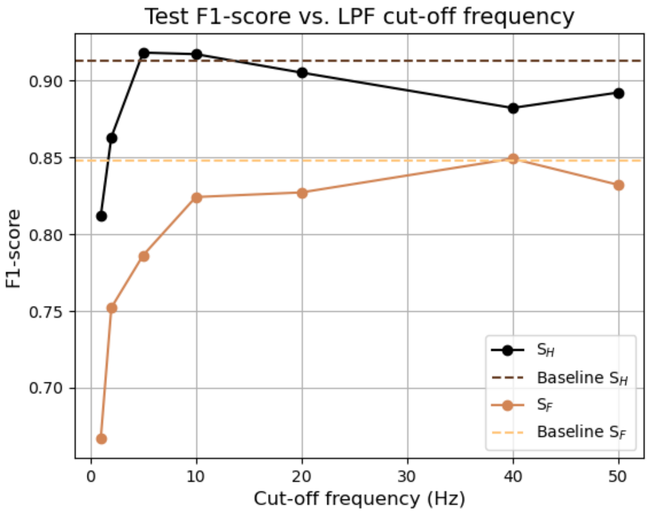


Fig. 9. Testing set F1-score versus low-pass filter cut-off frequency for both S_H and S_F

5.2.1 Time Gap & Window Size. To find out to what extent and how early the model can predict the different cycling maneuvers, multiple runs of the model are performed for various combinations of time gap and window size. For this evaluation, the raw 6 dimension IMU data from S_H are used as input features. Figure 8 shows the performance of the model in terms of F1-score for different combinations of time gap and window size W. It must be noted that when the time gap is increased the window is moved away from the maneuver earlier in time and that the turning maneuver is initiated at $G = 0$ seconds. The Figure shows that at a time gap of 0 seconds, all window sizes achieve an F1-score greater than 0.9. As the time gap increases it becomes increasingly more difficult for the model to correctly classify the classes. When the time gap is set to 1 second, all window sizes perform poorly with a less than 0.5 F1-score. It also seems that the bigger the window size the worse the F1-score across all shown time gaps. This might be because subtle movements indicate the maneuver and a bigger window size introduces more useless information to the model. It can be observed that the red line of $W = 1$ second lies the highest, meaning it achieves the best F1-scores for most time gaps. Moreover, it can be seen that the points at a 0.25 second time gap lie far apart. At this point in time, the 1 second window size performs significantly greater and already achieves an F1-score of 0.91. For this reason, W is fixed at 1 second and G is fixed at 0.25 seconds for the subsequent evaluations.

5.2.2 Comparing Sensors and Low-Pass Filtering. The amount of noise and what kind of movement is captured depend on the sensor placement on the bicycle. Through frequency analysis of both sensors, peaks were found around 15 Hz in the Fourier transforms of almost all dimensions of the IMUs. These are thought to be noise and it's not clear from the analysis which frequencies are present in cycling maneuvers. To compare the placements of S_H and S_F, raw and low-pass filtered performance is plotted in Figure 9 for both sensors. For this evaluation, the parameters found in Table 4 are kept constant. The low-pass filter is applied to all dimensions of the IMU sensor. In the Figure, the dotted lines *Baseline* S_H and *Baseline* S_F show the raw unfiltered F1-score of both sensor placements. It can be seen that S_H achieves a higher F1-score than S_F for all cut-off frequencies. In addition, there's a small improvement over the baseline performance of S_H at cut-offs of 5 and 10 Hz. The improvement is not significant and applying an LPF is an extra computation step and time is of the essence. Therefore, this improvement will not be included in the optimal model. Besides that, it's clear from the plot that there is a dip in performance below 10 Hz cut-offs. As the cut-off is moved closer to 0 Hz, more useful information is lost in the time window. It is observed that cycling maneuver indicators lie somewhere in the range of 0 to 10 Hz. Based on these results, the unfiltered S_H will be further evaluated in the following subsection.

5.2.3 Selecting and Processing Input Features. The next step is to find out which features of S_H contribute to classification performance to possibly decrease input data size without significant loss of performance. The same parameters as in the previous subsection are kept fixed for this evaluation. Figure 10 show the F1-score for removing individual dimensions of the raw S_H IMU, where the baseline performance is shown as the red dotted line. The biggest dip in F1-score is found by removing the x dimension of the gyroscope. Rotation around x is the tilt of the bike when the cyclist leans left or right, which turns out to contribute the most towards correct classifications. All features except accelerometer x and gyroscope y significantly help with classification. Additionally, the combination of removing both Acc X and Gyr Y is also investigated, which results in a high F1-score of 0.9. Consequently, both of these can be removed in the optimal model because it significantly decreases the input size.

Following that, several processed features are added individually to the raw 6 features of S_H as input. Figure 11 presents the F1-scores of adding angle, speed and the 6 features of S_F. A feature contributes to the performance if its bar crosses the red dotted line of baseline

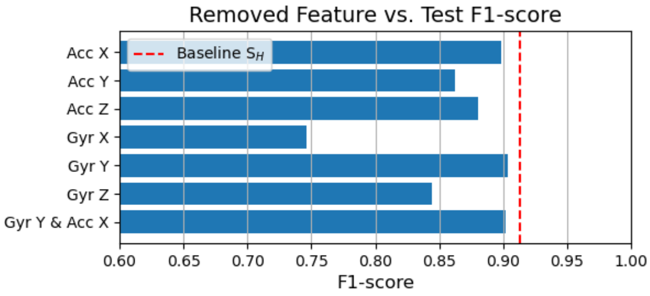


Fig. 10. Removed Feature of S_H IMU Versus Testing set F1-score

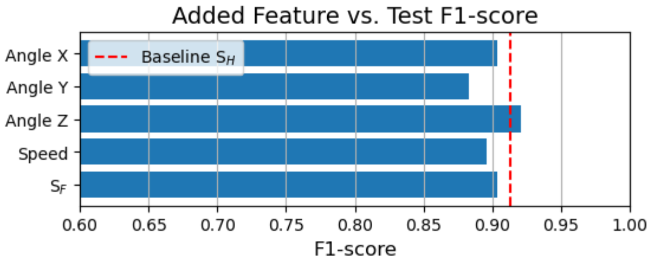


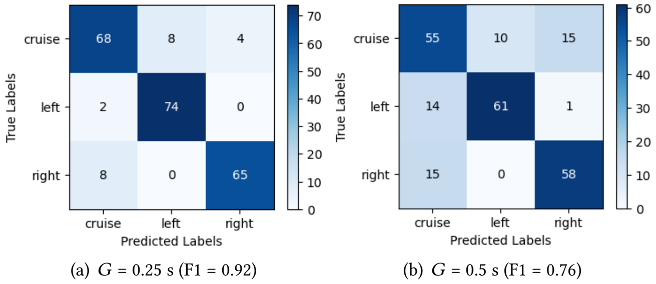
Fig. 11. Added Feature to raw S_H Versus Testing set F1-score

performance. The angle for a dimension is obtained by integrating the respective gyroscope dimension over time, using a cumulative sum function. From the three angles shown in the plot, only angle Z reaches a higher than baseline F1-score. A cumulative sum does not add significant processing time, so angle z will be included in the optimal model. Moreover, speed is obtained from the GPS, which is interpolated at 200 samples per second to match the input shape. The bar plot shows that speed does not increase F1-score. As a result, speed will not be included in the optimal model. Lastly, the plot also shows the performance of both S_H and S_F together, which is a total of 12 input features. Combining the sensors does not result in a better than baseline performance. In summary, for the optimal model the unfiltered features of accelerometer Y & Z gyroscope X & Z and angle Z of S_H are chosen.

5.2.4 Simplification of Convolutional Layers. In the previous section a selection of input features was made, which will be used for the evaluation in this subsection. Similar to previous evaluations, the same hyper-parameters are used that are found in Table 4. The goal of this evaluation is to find out whether the model can be simplified. Recall that Figure 5 shows the layered structure of the CNN-LSTM model. The number of filters in the first and second convolutional layer will be denoted C_1 and C_2 respectively. The default model has $C_1 = 16$ and $C_2 = 4$ with a total of 18952 trainable parameters. Table 5 displays the F1-scores obtained for different combinations of C_1 and C_2. The second layer was also removed and C_2 is then equal to 'None', which required also removing the connected MaxPooling layer. It can be seen that the F1-score roughly decreases as the first convolutional layer decreases. A maximum in F1-score can be found at $C_1 = 16$ and $C_2 = 2$ of 0.92, which is a slight increase over the default model. This configuration of the model counts 18214 parameters, a 738 parameter count difference with the default model. Consequently, for the optimal model these values for the number of filters is chosen.

Table 5. Number of filters in the first convolutional layer C_1 versus the number of filters in the second convolutional layer C_2.

C_2 \ C_1	16	8	4
4	0.91	0.91	0.87
2	0.92	0.89	0.81
None	0.88	0.85	0.88



(a) G = 0.25 s (F1 = 0.92)

(b) G = 0.5 s (F1 = 0.76)

Fig. 12. Confusion matrices of the final model for two different time gaps

5.3 Final Model

The final model consists of the following. The window size is set to 1 second and the features selected as input for the model are the accelerometer Y & Z gyroscope X & Z and angle Z of S_H. In the default model, the number of filters in the second convolutional layer is reduced to 2, which saves 738 parameters. The final model achieves a testing set F1-score of 0.92, 0.25 s before the start of the maneuver.

In addition, two confusion matrices are shown for time gaps of 0.25 and 0.5 seconds in Figure 12 to better understand how the final model performs. In the confusion matrix for $G = 0.25$ s we see a clear diagonal, indicating that most samples are being correctly predicted. From the top row it seems that samples labelled with cruise are sometimes confused with turning. This becomes more apparent for a bigger time gap, as shown in the confusion matrix for $G = 0.5$ s. A similar pattern is visible in the left most column in both matrices, it can be seen that both turning maneuvers are being confused with cruise. This makes sense, as the time gap increases less maneuver indicators are found in the time window, making it harder for the model to distinguish cruising and turning.

6 DISCUSSION

The evaluations show to what extent it is possible to predict a cycling maneuver. In this section the experiment limitations and obtained results will be discussed.

First of all, the experimental setup is a controlled environment. The intersection that was build using cones and chalk does not fully mimic a real world intersection. There are several factors present in real traffic that are not present in the setup that affect how cycling maneuvers are performed. Traffic signs, road markings, obstacles or other vehicles are some of these factors. In traffic a cyclist might have to swerve, slow down or make steering adjustments for certain scenarios, which can lead the model to think they are preparing for a turn. This means an increase in the number of false positives. Adding to that, not every road has the same texture, which can cause different kinds of noise. There can be experiments done involving different road types to take this into account. How well this setup reflects real world traffic could have been investigated by having the testing set consist of real traffic scenarios.

Secondly, from the results it's clear that the model somewhat confuses cruising with both turn directions for a time gap of 0.5 s. This misclassification is mostly likely to increase as the time gap increases, which is not ideal for a reliable prediction system. Section 7.1 proposes future work ideas to address this problem.

Thirdly, the time gap to predict is good but can be improved for more reliable maneuver prediction. The model is able to predict the maneuvers with an F1-score of 0.92 at 0.25 s before the start of the maneuver. To better prevent an accident, the prediction time gap could be increased. Also, it was not investigated how fast the model can predict when applied, which is a factor that further decreases the time to predict. Only two sensor placements were considered. Comparing more different kinds of placements could help increase the time gap. For example, a sensor on the pedals can show when a person stops pedalling. A different classification model could further reduce the time to process. Only the CNN-LSTM model architecture was evaluated. However, a simpler model architecture might be able to achieve similar or better performance. Besides that, the CNN-LSTM model was reduced in size, but not significantly. Other smaller neural network architectures with less trainable parameters can be researched.

Lastly, other definitions of the start of a turning maneuver are possible. This research used the definition that the turn maneuver starts when the angle of the handlebar is last aligned with the approach. If a different definition is used for the maneuver start, the results of this research would shift in time. It could be more accurate to extract t_m from another source than the motion sensor. A common method is to use a camera and manually choose when the maneuver occurs.

7 CONCLUSIONS

In conclusion, a CNN-LSTM model was used to predict the cycling maneuvers of turn left, turn right and cruise where the input is bicycle mounted motion sensor data in time windows. The model was able to predict 0.25 seconds in advance using a window size of 1 second with an F1-score of 0.92. It was found that sensor placement on the handlebar is more effective than placement on the frame, where a low-pass filter does not significantly improve either of the placements. The Y dimension of the gyroscope and X dimension of the accelerometer can be left out of the input and angle Z can be added to the input. In addition, there are some limitations to the method and model used. An experimental setup was build with cones instead of a real traffic environment. Also, with the current sensor placement, as the time gap increases it becomes increasingly more difficult for the model to distinguish cruising and turning.

7.1 Future Works

There are several potential follow up studies that can improve or add to this research. First of all to reduce misclassification, more cycling maneuvers or subtler movements can be added to the model. For a maneuver prediction model to be implemented in real traffic, it needs to be able to deal with the wide amount of cycling maneuvers. Maneuvers such as changing lanes, overtaking, braking and crossing an intersection were not investigated in this work and can be added to the model. In these maneuvers, subtle movements may be telling and could be researched in more detail. It's imperative to make a distinction between predicting a maneuver and detecting a maneuver.

Adding to that, insights from existing research can be combined into one big cycling maneuver prediction and monitoring model. Adding more sensors could increase the prediction time gap and improve classification performance. In traffic, it's common practice to indicate your turn direction using your hand. As mentioned in the related work section, DoubleCheck proved it's possible to detect single-hand cycling with a smartphone on the handlebar. This insight can be added to a maneuver prediction model. Additionally, dangerous riding behaviour detection by BikeMate such as wrong-way cycling or head movement like in HeadMon can be added to the model. Alternatively, undiscovered behavioural models using motion sensor can be investigated such as detecting drunkenness, tiredness or nervousness of a cyclist in traffic. This could also lay a basis for tracking health characteristics of a cyclist or even behaviourally dependent motorised assistance on bicycles.

Lastly, there has not been much research into applying maneuver prediction models in traffic. Active protection systems can be designed to use such a model to intervene by braking, steering or notifying the cyclist. One potential application of maneuver prediction is by alerting other road users. Development of Vehicle-to-Vehicle (V2V) technology has become possible, including for bicycles [11]. By using V2V, a road user's maneuvers can be broadcast in real-time. Surrounding vehicles can receive the maneuver intention of other road users and take action.

REFERENCES

[1] M. van Algemene Zaken, "Cycling facts 2018," Apr 2018. [Online]. Available: https://www.government.nl/documents/reports/2018/04/01/cycling-facts-2018

[2] C. B. voor de Statistiek, "Centraal bureau voor de statistiek," Apr 2023. [Online]. Available: https://www-cbs-nl.ezproxy2.utwente.nl/nl-nl/nieuws/2023/16/meer-verkeersdoden-in-2022-vooral-fietsende-75-plussers-vaker-slachtoffer

[3] W. Gu, Z. Zhou, Y. Zhou, H. Zou, Y. Liu, C. J. Spanos, and L. Zhang, "Bikemate: Bike riding behavior monitoring with smartphones," in *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, ser. MobiQuitous 2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 313–322. [Online]. Available: https://doi-org.ezproxy2.utwente.nl/10.1145/3144457.3144462

[4] Z. Han, L. Xu, X. Dong, Y. Nishiyama, and K. Sezaki, "Headmon: Head dynamics enabled riding maneuver prediction," in *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2023, pp. 22–31.

[5] S. Gupta, R. V, and S. A, "Fall accident detection system for bicycle riders using support vector machines," in *2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT)*, 2022, pp. 392–397.

[6] N. Wijerathne, S. K. Viswanath, M. S. Hasala, V. Beltran, C. Yuen, and H. B. Lim, "Towards comfortable cycling: A practical approach to monitor the conditions in cycling paths," *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018.

[7] Y. Usami, K. Ishikawa, T. Takayama, M. Yanagisawa, and N. Togawa, “Bicycle behavior recognition using sensors equipped with smartphone,” in *2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, 2018, pp. 1–6.

[8] R. Benitah, “Cyclist weight inference from bicycle-mounted sensor data,” February 2023. [Online]. Available: http://essay.utwente.nl/94344/

[9] X. Dong, Z. Han, Y. Nishiyama, and K. Sezaki, “Doublecheck: Detecting single-hand cycling with inertial measurement unit of smartphone,” in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2022, pp. 50–53.

[10] J. R. Galli and B. W. Carroll, “The Four-Ball Gyro and Motorcycle Countersteering,” *The Physics Teacher*, vol. 55, no. 4, pp. 238–239, 04 2017. [Online]. Available: https://doi.org/10.1119/1.4978726

[11] S. Blanco, “Ces 2021: Bicycles want to get into the connected vehicle game,” Jan 2021. [Online]. Available: https://www.forbes.com/sites/sebastianblanco/2021/01/15/ces-2021-bicycles-want-to-get-into-the-connected-vehicle-game/