ABSTRACT

Cycling is a common mode of transport in the Netherlands, however there is also a high amount of cyclist injuries every year. Identifying and improving the safety of unsafe situations on the road can reduce the number of injuries. Previous studies have used historical data to identify these situations. This paper explores a proactive approach to identifying these situations before crashes happen. In order to accomplish this, deceleration rate is used as a surrogate safety measure.

Keywords
Detection, Surrogate Safety Measures, Cycling, Smartphones

1. INTRODUCTION

Cyclists account for a significant portion of traffic injuries in the Netherlands. While merely eight percent of traveled kilometers are made by bike, cyclists account for 31 percent of traffic related deaths [1]. Previous studies on cyclist safety have used crash and injury rate in order to identify the safety impacts of particular sections of the road [3]. A limitation of this approach is that it is reactive: it relies on historical data, requiring crashes to happen first. Ideally dangerous sites should be identified as such as soon as possible, instead of having to wait several years until a large number of accidents has already happened. This paper explores a method to identifying these dangerous sites proactively by making use of smartphone sensor data. Several safety measures exist to recognize the dangers of a situation. These safety measures include, but are not limited to, time-to-collision, post-encroachment time, gap time and deceleration rate [7]. For the purposes of this research, hard braking events are used as a surrogate safety measure. Previous research has shown that the frequency of hard braking is correlated with crash occurrence for vehicles. This is because hard braking indicates an evasive action to a possible accident. To conclude, this research focuses on identifying dangerous sites by using a smartphone accelerometer.

2. PROBLEM STATEMENT

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

A large portion of traffic related injuries and deaths are from cyclists. Many of these accidents happen at hotspots, sections of road that are more dangerous than others. Although some research has been conducted to identify these hotspots, no effective method exists to map an entire road network. Previous research often required historical data in order to draw conclusions [3] and was otherwise not yet reliable enough to use [7]. Therefore, more research into using surrogate safety measures to reliably determine unsafe situations on the road is needed to ensure a safe traffic environment for cyclists in the Netherlands.

3. RESEARCH QUESTIONS

The research addresses the following research question: How accurately can smartphone sensor data from cyclists be used to recognize unsafe situations on the road?

In order to logically derive to a complete answer to this question, this paper aims at answering several sub-questions.

1. What types of cycling events can be used as surrogate safety measures?
2. How accurately can these be detected using smartphone data?
3. How can these events be mapped to the actual location?

4. BACKGROUND

4.1 Surrogate Safety Measures

For several decades surrogate safety measures have been used as road safety indicators. While a vast amount of surrogate safety measures exists, the "best" are gap time, time to collision, post encroachment time and deceleration rate [4].

**Definition 4.1. Gap Time (GT) Gap Time is the time between completion of encroachment by the first vehicle and the arrival time of the second vehicle if they both continue with the same speed and path.**

**Definition 4.2. Time-to-collision (TTC) Expected time until two vehicles collide if they remain at their present speed and on the same path.**

**Definition 4.3. Post-Encroachment Time (PET) Time difference between lead vehicle entry to a collision point and the arrival of a following vehicles at the same point.**

**Definition 4.4. Deceleration Rate Rate at which vehicle must decelerate to avoid collision. This is considered an appropriate measure for the detection of dangerous driving maneuvers.**
4.2 Dynamic Time Warping

Dynamic Time Warping is an algorithm that can be used to measure the similarity between two signals which may vary in speed[2]. The input of the algorithm is two series, one unknown series $S$ and a template series, $T$.

$$S = s_1, s_2, s_i, s_n$$

$$T = t_1, t_2, t_i, t_m$$

Using these two series an $n$-by-$m$ matrix can be constructed, where each element $(i, j)$ contains the distance between the two points $s_i$ and $t_j$, where the distance is

$$d(s_i, t_j) = (s_i - t_j)^2$$

A warping path, $W$, maps the elements of $S$ and $T$ such that the distance between them is minimized.

$$W = w_1, w_2, ..., w_k$$

Here the $k^{th}$ element is denoted as $w_k = (i, j)_k$. Next the minimum distances along the warping path need to be summed in order to find the cost $C$.

$$C(X, Y) = \sum_{k=1}^{n} w_k(x_{nk}, y_{mk})$$

The template with the lowest cost $C$ is the best match for the input signal. This implementation of DTW runs in $O(n^2)$ time. Faster algorithms have been developed, but have the downside that they do not always find the optimal warping path [6].

5. RELATED WORK


Salvador and Chan [6] present a faster DTW algorithm. Improving the complexity of the algorithm from $O(n^2)$ to linear time, allowing the algorithm to be used on much larger data sets.


Strauss et al. [7] use deceleration rate as a surrogate safety measure to identify unsafe situations for cyclists. Only GPS data is used in order to achieve this. A rule-based algorithm is used where any deceleration above 3.4 $m/s^2$ is considered as dangerous. A top 10 of most dangerous corridors was made based on both historical data and the dangerous deceleration rate. The results were compared with historical accident data. Six of the top ten were the same based on both criteria, showing that this method can be used to identify dangerous locations. The low frequency of GPS limits the effectiveness of the algorithm.

Saiprasert et al. [5] use smartphone sensor data to detect driving events. Three algorithms are compared. The first algorithm uses GPS data and a rule based algorithm, the second uses accelerometer data and pattern matching. DTW is used to pattern match the data with known driving events. The third algorithm is an improvement on the second to reduce the required computational power. The pattern matching algorithm is found to be significantly more effective than the rule-based algorithm.

6. METHODOLOGY

Since this research will only use smartphone sensors as input data, the majority of the mentioned surrogate safety measures cannot be used. However the existing research does prove that detecting evasive actions can be used to assess the dangers of a situation. This research will focus on identifying hard braking actions using accelerometer data. The research by Strauss et al. used GPS to detect these events, but using an unreliable GPS signal for such measurements decreases the effectiveness of the algorithm[7].

6.1 Data collection

This section describes the platform that is used to collect the data for the detection of braking events. For this research, two sensors from an Android smartphone are used. Namely, the 3-axis accelerometer and the GPS receiver. The accelerometer measures acceleration, caused both by the phone’s movement as well as gravity. The three axes correspond to lateral, longitudinal and vertical accelerations. Figure 1 shows the accelerometer axes in reference to the phone. Lateral movement is denoted by the x-axis and the longitudinal movement is denoted by the y-axis.

A pre-existing Android application was used to collect and save the data. With this application, the GPS receivers’ data is sampled with a frequency of 1Hz. The accelerometer is sampled with a rate of 80Hz.

6.2 Pattern matching algorithms

The pattern matching algorithm consists of several steps. First, to make the raw sensor data usable, it is preprocessed to reduce unwanted noise. Next, an event recognition algorithm is applied to find the sections in the data where a driving event may occur.

Two different classification algorithms will be tested. The first is a simple rule-based matching algorithm and the second is a pattern matching algorithm that uses dynamic time warping. For the rule based algorithm constraints need to be set to classify the events. For the pattern matching algorithm reference patterns need to be created. This is done by labeling preprocessed data of which the events are known. These reference patterns can then be compared with data of which the event is not known to find the matching event. Figure 2 shows an overview of the steps of the pattern matching algorithm.

The preprocessing step, the event recognition and the event classification steps have all had two different algorithms implemented for them. All combinations of these algo-
7. PREPROCESSING
To find and properly classify driving events in the gathered data several steps need to be performed. First the data needs to be preprocessed to make it usable. The raw gathered data contains a large amount of noise caused by the vibrations in the bicycle. This kind of noise is expected for any kind of moving vehicle, but is significantly more severe for a bike than a vehicle such as a car. This is caused by the fact that bikes do not have a suspension, so even the slightest bump in the road is transferred directly to the phone. Two methods were tested to find the most effective smoothing filter.

7.1 Moving average
A simple moving average (SMA) can be used to smooth out the spikes in the data [9]. By changing the number of data points that are used to calculate an average with, the intensity of the smoothing can be modified. Riding the bike at higher speeds or riding over a bumpy road causes the spikes in the data to be bigger. To combat this the number of data points that contribute to the average has to be increased, but this has the unwanted side effect of reducing the intensity of braking events as well. Especially hard braking events with shorter peaks are smoothed out retaining the intensity of braking events than the simple moving average.

7.2 Exponential moving average
An exponential moving average (EMA) applies weighting factors to the data points which decrease exponentially [9]. The following formula is used to calculate the EMA:

\[ S_t = \begin{cases} Y_t, & t = 1 \\ \alpha Y_t + (1 - \alpha) S_{t-1}, & t > 1 \end{cases} \]

Here the value of \( \alpha \) can be used to modify the degree of weighting decrease. A lower \( \alpha \) will smooth out the data more. \( Y_t \) is the value of the data point at time period \( t \) and \( S_t \) is the value of the EMA at time period \( t \).

This method is more effective at smoothing out the noise whilst keeping the intensity of braking events than the simple moving average.

7.3 Reducing the number of data points
Next, the number of data points of the raw data is significantly reduced. The input data has almost 80 data points per second, which is unnecessarily high and causes longer processing times in later steps of the algorithm. The data is reshaped to have 4 data points per second.

7.4 Finding the corresponding GPS location
To get the location that corresponds with an accelerometer data point, the entries of two separate data sets need to be matched. First, the GPS data needs to be checked for missing data. Due to signal losses there can sometimes be several seconds where there is no GPS location. This missing data is added by simply repeating the last known location for the duration of the signal loss. Since the signal losses are generally short this is a good enough solution. Although the start time of the two data sets can be different, the end time is always the same. Therefore by calculating back from the end of the file the closest GPS location can be found for every accelerometer data point.

The preprocessed accelerometer data and their matching GPS points can then be saved as one data set, so that the location of a data point can easily be accessed in later steps of the algorithm.

8. EVENT DETECTION
Saiprasert et al. [5] propose an extra step that can increase the robustness and efficiency of the algorithm. This step identifies where in the data events occur before applying the relatively slow DTW algorithm. Since this step proved to be very useful for event detection on data that was collected in a car, the same will be attempted for this research. Just as with the preprocessing step, several different algorithms have been tested to find the most effective one.

8.1 Detection using standard deviation
In this approach a moving standard deviation is used as a threshold, which can be compared with the standard deviation of the current window. If the standard deviation of the current window is greater than the threshold then the rest of the algorithm will be executed. The standard deviation of the window will only be bigger than the threshold if the spread of the acceleration data is sufficiently high. Such a high spread indicates that a driving event occurs in the window.

The threshold standard deviation is calculated with every new data point using the following formula:

\[ \sigma_{th} = \sqrt{\frac{(i - 2)\sigma_{th_{i-1}}^2 + (i - 1)(\mu_{i-1} - \mu_i)^2 + (a_i - \mu_i)^2}{i - 1}} \]  

Where \( i \) denotes the current data point being processed, \( \sigma_{th} \) denotes the standard deviation of the accelerometer data up to the current point, \( \mu_i \) is the mean of the accelerometer data from the start up to the current point and \( a_i \) denotes the current accelerometer value.
Even though the preprocessing step is used to remove the vibration noise from the data as much as possible, there is still a significant amount of noise in the data. In areas where the road contains many bumps the standard deviation will be a lot higher. This causes these areas to possibly be wrongly identified as events. Additionally these sections cause the threshold to rise higher than the standard deviation of an actual braking event on a smooth road. The effectiveness of using standard deviation as event detection is thus very dependent on the amount of noise in the data.

8.2 Detection using mean
Since braking events are the only driving event that is interesting for this research, the values on the Y-axis of the accelerometer will always be lower during an interesting event. So instead of using standard deviation to detect events a much simpler method can be used.

8.2.1 Detection using mean
With every incoming data point a mean of the total data set is updated using the following formula.

\[ m_n = \frac{(n-1)m_{n-1} + a_n}{n} \]

(2)

Although the mean of the total set should be close to zero, this is not always the case. If the smartphone is not completely level, then gravity will cause the value on the Y-axis to be shifted.

The mean of the data set up to the current point is compared with the current window. If the mean of the current window is significantly lower than the average then the current window is marked as a possible event.

9. EVENT CLASSIFICATION

9.1 Dynamic Time Warping
The previous step produces a list of possible events. These events are then compared to our reference data using the dynamic time warping algorithm. The distance between the current event and all reference events is calculated for both event types. If a low enough distance is found, then the event type with the lowest distance is chosen as the type.

9.2 Deceleration Threshold
A second classification algorithm will be implemented, so that it can be compared with the effectiveness of the DTW algorithm. This method uses several threshold values on the Y-axis data to determine the event type. The mean of the whole data set is compared to the current accelerometer value, if the current value is significantly lower than the mean then it is likely that a braking event has occurred.

10. MAPPING EVENTS TO ROAD SECTIONS
As a final step, the classified events now need to be matched to sections of road. These sections are either an intersection, or the section of road between two intersections. Any event within a 30 meter radius of an intersection is considered to belong to that intersection. To find the distance between two coordinates the Haversine formula is used [8].

\[ h = 0.5 \times r \times \arcsin \left( \sin \left( \frac{\varphi_2 - \varphi_1}{2} \right) + \cos \varphi_1 \times \cos \varphi_2 \times \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right) \right) \]

(3)

where

- \( h \) is the distance between the two points
- \( r \) is the radius of the earth
- \( \varphi_1, \varphi_2 \) are the latitude of point 1 and latitude of point 2, in radians
- \( \lambda_1, \lambda_2 \) are the longitude of point 1 and longitude of point 2, in radians

Next, solve for \( d \) by using the arcsine function:

\[ d = 2r \arcsin \left( \sqrt{h} \right) \]

(4)

Using this, the distance to all known intersections can be calculated. Some event locations are not within the radius of an intersection. If this is the case, then the road section where the distance to the start and end intersection is the lowest is chosen as a match.

11. EXPERIMENTAL SETUP
To answer the research questions, two separate experiments have been performed. Both experiments use the same method of collecting the data. For the data to be usable for this research the axes of the phone always need to be in the same direction with respect to the bike. To achieve this the phone is secured to the back of the bike, with the top side of the phone pointing towards the front of the bike. With this setup the Y-axis of the accelerometer is an acceleration/deceleration, the X-axis is a sideways motion and the Z-axis is an up or down motion.

In the first experiment data was collected by one cyclist on one bicycle. The cyclist was instructed to ride through a city center just as he would normally, whilst saying it out loud whenever a braking action was performed. Sound was recorded during the experiment using an extra telephone, this makes it possible to label all events after the experiment was completed. In total, 37 braking and 35 hard braking events have been labeled. For the braking event, 27 are used as training data, while for the hard braking event 25 are used as training data. This leaves 10 events remaining to be used as test data for both types of events.

In the second experiment raw data was collected by three different cyclists on three different bikes. This means the data contains different types of cyclists which may have a different style of riding, as well as different types of bikes that have different vibrations and braking strength.

A route running through the city center was chosen before the experiment was carried out. This route contains several busy and dangerous intersections as well as some intersections that are more quiet and safe. The route also contains different types of pavement, with a portion of the route being on a brick paved road which contains more bumps than smooth asphalt. Figure 3 shows the route that was taken. Dangerous intersections are marked in red and pedestrian areas are marked in blue. For each of the three cyclists around 80 minutes of data was collected. During this time the route was completed 18 times by each of the cyclists.

12. EXPERIMENTAL EVALUATION

12.1 Preprocessing algorithm
Figure 4 shows a braking event in the raw data. Due to the noise in the data it is difficult to see the braking event. Note that the y-axis of this graph is in m/s², so the spikes in the data go far beyond the type of acceleration or deceleration that a cyclist can realistically perform.

The same braking event is shown in figure 5, this time with smoothing applied to it. Here the exponential moving
average algorithm was used with an $\alpha$ of 0.01. Now that the noise has been mostly smoothed out of the data it is clearly visible that a braking event occurred. Due to the smoothing the $m/s^2$ deceleration that is shown in the graph is now lower than the real deceleration of the event.

The need for a preprocessing algorithm is clear, but a comparison needs to be made to find the most effective algorithm and settings. To accomplish this two measures are used to compare the effect of preprocessing on the effectiveness of the complete algorithm. The first is the true positive rate (TPR), calculated using formula 5.

$$TPR = \frac{TP}{P}$$ (5)

Here $TP$ stands for the number of true positives, equivalent with a hit. $P$ stands for the total number of positive cases in the data.

However, just because an algorithm detected all events does not necessarily mean it actually performed well because it might have be making many false positive detections as well. For this, a measure called false discovery rate (FDR) was used. This is calculated using formula 6.

$$FDR = \frac{FP}{FP + TP}$$ (6)

Here $FP$ stands for the number of false positives and $TP$ stands for the number of true positives, just as in the above formula.

The results of this can be seen in figure 7. In the first results column, it can be seen that applying the algorithms without smoothing the data produces no useful results. The raw data contains so many vibrations that they completely drown out any braking event.

As explained during the preprocessing section, there is a ‘sweet spot’ where the preprocessing is the most effective. Running the algorithm with less aggressive preprocessing will leave more noise, while running the algorithm with too aggressive preprocessing will smooth out events so much that they become less distinguishable. This expected effect can be seen in the results. For example when using the simple moving average for smoothing, the algorithm performs better with $n = 120$, than with $n = 80$ or $n = 160$. The same is true for the exponential moving average, with $\alpha = 0.01$ having the best performance.

During preprocessing the number of data samples per second is reduced. This is done to reduce the execution time of the algorithm. Reducing the number of samples per second by too much will make events less recognizable, while on the other hand leaving the samples per second high will make the algorithm unnecessarily slow. Figure 7 shows the effect of the samples per second on the classification algorithm.

As can be seen in the table, leaving the samples per second higher than 4 has practically no influence on the effectiveness of the classification algorithm. Lowering the number of samples even more does have a clear negative impact, therefore the rest of the research will use data with 4 samples per second.

12.2 Detection and classification algorithm

Figure 8 shows the confusion matrices of the different combinations of algorithms. For both the braking and the hard braking event there are 10 events that should be detected by the algorithm. If an algorithm performs perfectly, then
only the green diagonals would be filled in. For all the algorithms discussed here, the data was preprocessed using the exponential mean average with an $\alpha$ of 0.01.

The first test uses mean for event detection and the dynamic time warping algorithm for classification. The algorithm correctly identified 40% of braking events and 80% of hard braking events. The other two hard brake events were incorrectly identified as normal brake events, additionally one normal brake event was incorrectly identified as a hard brake event.

The second test uses standard deviation for event detection and once again the DTW algorithm for classification. The effectiveness of this algorithm is severely limited by the large amount of vibration noise present in the raw data. Although the preprocessing step attempts to remove this as much as possible, it still remains problematic. In a part of the data where the bicycle goes over a bumpy road the standard deviation threshold will become higher due to the increased noise. This threshold becomes so high that some hard braking events no longer pass this threshold.

The third test uses mean for event detection and the simpler rule-based classification algorithm. This algorithm actually performs better when it comes to detecting normal braking events, but detects less hard braking events.

The fourth test uses standard deviation for event detection and the rule-based classification algorithm. Just as with the DTW tests, this version performs worse than the version that uses mean as detection.

Overall, the first algorithm has the best performance. Since the correct classification of hard braking events is much more important than the correct classification of normal braking events for this research, it outperforms the three other algorithms.

### 12.3 Mapping to actual locations

Now that the most effective version of the algorithm has been determined, it is time to address the main research question of this paper. In order to answer this the events are mapped to their corresponding road section. The division into sections can be seen in figure 9. The number of hard braking events that fall into a section are counted and the results are shown in figure 10. As can be seen in this table, there are three intersections that stand out due to their number of hard braking events. These intersections are numbered 1, 2 and 6. All these intersections are
busy and cluttered, so this is the expected result. Besides these intersections there are also a few road sections that have a high amount of hard braking events. The sections between intersections 2 and 3 and between intersections 3 and 4 run through a busy pedestrian area, so the high amount of hard braking events here was also expected. However between intersections 7 and 8 there are also a number of hard braking events. Unlike the other sections, this section is not busy or cluttered, so this section should not have such a high amount of detected hard braking events.

13. CONCLUSION

The algorithm found all the areas that, before the experiment was performed, were expected to be dangerous. However there was also a safe road section that the algorithm considered dangerous. Both in terms of detection rate as well as false detection rate, significant improvements need to be made before the algorithm can be used to base important infrastructure decisions on.

There are two problems that significantly reduce the effectiveness of the algorithm. The first of these problems is that the quality of the collected data is low, with a high amount of noise present in the raw data. In future research a mechanical solution for reducing this noise should be explored. Preprocessing steps can be used to reduce noise, but they also reduce the differences between braking events. Reducing the vibrations measured by the phone by mounting it on a shock absorbing material such as rubber or foam may be a simple solution to this problem. The second problem is that there is often a small difference between a braking and a hard braking event. This same issue was also encountered in the research by Saiprasert et al.[5]. In their research a large amount of different driving events could be effectively classified, but distinguishing hard braking events from braking events proved to be the most difficult. One simple improvement that can be made over the current experiment is to use bicycles that are in a better condition. The brakes on the bicycles were in a poor condition, making it impossible to brake really hard.

Having better brakes could make the differences between a braking and a hard braking event bigger, thus making it easier to correctly classify events.

In conclusion, this method of road analysis shows promise, but more research needs to be done into improving the effectiveness of the classification algorithm. Also, more research needs to be done into how well the results of this algorithm match up with historical accident data. Although hard braking events have been proven to be usable as a surrogate safety measure for cars, the effectiveness of this for cyclists is mostly unknown.

14. REFERENCES


