

# DANGER DETECTION FOR CYCLISTS WITH MACHINE LEARNING (IN THE CITY OF COPENHAGEN)

Mara Alena Lehmann<sup>1</sup>, Daniel Pascal Mair<sup>2</sup>, Gabriele Stefanie Gühring<sup>3</sup>

<sup>1,2,3</sup> Hochschule Esslingen – University of Applied Sciences, Flandernstr. 101, 73732 Esslingen am Neckar, Germany

Received 28 February 2022; accepted 20 March 2022

**Abstract:** This paper offers several ways to classify time series data recorded by cyclists in an urban area like Copenhagen to predict and classify dangerous situations and areas. Therefore, several neural networks used a training dataset of bicycle trips consisting of position data and associated system modes derived from a Support Vector Machine. The system modes indicate if cyclists are in dangerous situations. The model used position data and derived features like velocity, acceleration, angular deviation, and the deviation of the previous cycling behaviour in the respective trip. A gated recurrent neural network model achieved the best resulting accuracy of 83 % in a binary classification between accident and no danger. Through this, it was possible to determine if a bicycle accident happened due to the cyclist's environment e.g., cobblestones, or due to their cycling behaviour. This way the dataset and the approved machine learning model can show municipality of cities which spots are currently posing a threat for cyclists. Furthermore, the developed algorithm can pose as a basis for a cyclist app that warns its user about dangerous driving behaviour or upcoming danger spots. All the developed algorithms can be transformed to other cities.

**Keywords:** accident analysis, bicycle position data, danger classification, neural network, trip segmentation.

## 1. Introduction

In recent years, the bicycle has become more and more popular not only as a recreational device, but also as an everyday vehicle, especially in cities. On the one hand, it convinces with its practicality and its price compared to the car, on the other hand, it is an emission-free vehicle, which is important for meeting the climate goal of the global community. Therefore, it is not surprising that an increasing number of cities are shifting their focus away from the car as the mobility solution for individuals and expanding alternative modes of mobility such as the bicycle.

However, with the increased traffic volume of cyclists, the number of traffic accidents involving cyclists is also increasing (International Transport Forum, 2013). Furthermore, many collisions involving cyclists are not reported and therefore cannot be recorded in statistics, so the dark figure is likely to be a lot higher (International Transport Forum, 2013; Watson, Watson, & Vallmuur, 2015). Hence projects like SimRa (Karakaya, Hasenburg, & Bermbach, 2020) exist that make it possible for cyclists to report an accident or a dangerous situation during a bicycle trip. With this data they identify dangerous locations within cities.

<sup>1</sup> Corresponding author: maleit11@hs-esslingen.de

To ensure the safety of cyclists, many cities like Copenhagen (Cycling Embassy of Denmark, 2018) invest money in the prevention of cycling accidents and general road safety. The Copenhagen City Council has set a target to reduce the number of serious injuries or fatalities on the roads to zero by 2025. Values from 2017 report 117 serious injuries or fatalities in Copenhagen, with cyclists or pedestrians involved in 70% of the cases.

This paper took up and expanded on the point of accident prevention for cyclists by using machine learning methods with time series data of cyclists in Copenhagen. The training dataset consisted out of position data plus calculated system modes of bicycle trips collected with the helmet of the company Hövding in Sweden. The system modes describe with the help of a Support Vector Machine (SVM) (Steinwart & Christmann, 2008) if a trip is in a dangerous situation (Lindqvist & Roos, 2020). By analysing parts of the time series, hereafter referred to as *trip segments*, it was possible to identify areas that have an increased risk of accidents. The driving behaviour of trip segments that end in accidents are crucial for such an investigation. Different neural networks from the field of supervised learning were used as analysis tools, which recognize the different reasons for accidents through pattern recognition. This paper stands out in the fact that it analyses trip segments from cyclists to draw conclusions if their driving behaviour leads to an accident or the environment of the location the accident happens.

This research presents for municipalities of cities the opportunity to show places that are

currently posing as a threat for cyclists. The results of this paper will be used to develop a mobile app that is able to warn cyclists about dangerous places as well as dangerous cycling behaviour.

## 2. Related Works

Analysing position data trajectories with various machine learning methods to predict traffic flow or driving behaviour is part of several papers over the last few years.

Using taxi position data trajectories, Wang *et al.* (2018) investigate behavioural patterns of taxi drivers in Chinese cities. Their aim is to detect taxi fraud by identifying conspicuous taxi routes through hierarchical clustering. Taxi fraud happens when a taxi takes a longer route than usual in order to earn the driver more money. To detect this, the authors compare how often certain driving decisions are made compared to others on similar trajectories. In addition to Global Positioning System (GPS) data, they also use video data. Their resulting algorithm enables them to automatically detect four conspicuous patterns of behaviour among taxi drivers and thus uncover possible cases of fraud.

Jiang *et al.* (2017) present a Recurrent Neural Network (RNN) for identifying means of transportation based on position data trajectories. They distinguish between four categories of transportation: walking, cycling, bus and car. The RNN achieves 98 % accuracy using point-based and segment-based features such as speed and average speed of the segment. The dataset consists of approximately 2.2 million GPS records in Beijing, China.

A similar paper from Dabiri *et al.* (2018) achieves a high accuracy for identifying means of transportation based on raw GPS trajectories with a Convolutional Neural Network (CNN). In addition to walking, cycling, bus and car they also identify if the GPS trajectories originate from a person on a train. They use a CNN, which is common in computer vision (Krizhevsky, Sutskever, & Hinton, 2017), due to its ability to find local patterns.

In (Saiprasert, Pholprasit, & Thajchayapong, 2017) the authors present different approaches to identify driving events such as breaking or turning. Their data does not come from devices installed in the vehicle, but from smartphones of the passengers. They use speed, GPS position and cardinal direction as input for their algorithms. The algorithms enable them to distinguish sudden driving manoeuvres from normal driving manoeuvres. Consequently, it is possible to deduce whether there is a potentially aggressive driving behaviour.

In addition, Holmgren *et al.* (2020) use position data trajectories to identify unsafe places for cyclists within the city of Lund in Sweden. Their work is close to the present paper in that they analyse bike data and use a Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm as presented in (Ester, Kriegel, Sander, & Xu, 1996). Their dataset consists of position data points that are classified as unsafe by cyclists themselves and not by automatically calculated value of an SVM. While cycling, cyclists have the possibility to manually record an unsafe place by pressing a button on the handlebar. The current

GPS position and time is then stored. The data is analysed by the clustering methods k-means (MacQueen, 1967) and DBSCAN. Holmgren *et al.* (2020) thus detect several clusters within Lund that are classified as unsafe for cyclists. However, the present work uses position data recorded by cyclists wearing the airbag helmet of Hövding. It classifies each position data point with a system mode via an SVM. The system mode indicates dangerous situations.

The company Hövding in Sweden produces a special kind of bicycle helmet, which consist of an airbag that elevates in case of an accident. As part of the latest version *Hövding 3*, the helmet connects with the user's smartphone which then collects data from the helmet. Besides the position data and timestamp the helmet uses a system mode which calculates if there is an accident or a low, medium or high danger situation for the cyclist.

The dataset of Hövding is subject of a previous study by Lindqvist and Roos (2020). They research hazard clusters in the city of Malmö in Sweden by using DBSCAN. Comparing the clusters determined by DBSCAN with accident data provided by the Swedish Traffic Accident Data Acquisition (STRADA, 2021) most accidents detected by the Hövding helmet and registered by STRADA corresponded to a category which did not include fixed barriers such as speed bumps or cobblestones nor construction work and therefore could not be explained by external influences.

The present paper fills this gap and distinguishes for each hazard cluster whether

it belongs to a dangerous environment or a dangerous driving behaviour. RNNs and CNNs (Dabiri & Heaslip, 2018; Sun, Chen, & Sun, 2019) are used to classify each trajectory running through a danger cluster detected by the Hövding helmet. In contrast to Lindqvist and Roos (2020), this paper uses a dataset collected in the city of Copenhagen in Denmark. However, after collecting the necessary amount of bicycle trip data all the developed algorithms of this paper can be transformed onto other cities.

### 3. Data and Data Preprocessing

#### 3.1. Data

The geographical position of the bicycle data used consists of recorded data of latitude, longitude and altitude, a timestamp and current status or system mode which describes if a trip is in a dangerous situation (see Table 1). The helmet records the data with sensors, collects and stores it in JSON trip files. The data of each trip belongs to one pseudonymized user.

	user	trip	altitude	latitude	longitude	timestamp	systemMode
...	...	...	...	...	...	...	...
7	1	30	15.331279	55.667553	12.533282	2019-10-28 06:24:52.789	4
8	1	30	9.193001	55.667523	12.531149	2019-10-28 06:24:53.492	4
9	1	30	8.213308	55.667547	12.531015	2019-10-28 06:24:56.182	1
...	...	...	...	...	...	...	...
38460	983	88	49.500000	55.401676	10.425285	2020-04-28 12:27:39.348	1
38461	983	88	49.500000	55.401680	10.425449	2020-04-28 12:27:42.415	1
38462	983	88	49.500000	55.401631	10.425458	2020-04-28 12:27:51.000	2

**Fig. 1.**

*Multidimensional Trip Data*

*Note: A multidimensional trip consists of altitude, latitude, longitude, timestamp and a system mode which describes the danger status of a data point.*

For the current status of each position in the trip, a SVM embedded in the bicycle helmet calculates the system mode which can be used for classification (Steinwart & Christmann, 2008). System modes 2 to 6 represent a danger status and are the central component for danger classification (see Table 1). Accident trips end with system mode 2, then the helmet is inflated. Table 1 describes the original system modes determined by the SVM of the Hövding helmet in the first two

rows. The third row shows the mapping of this data for the purpose of this paper so that only four relevant classes are left. As the helmet records system modes 2 to 6 in trips which also contain system modes 8 and 9 at some position points, those can be mapped to system mode 1 which represents no danger. This study deletes data points with system modes 0, 6 and 7 because they are lacking relevant information and are very seldomly recorded.

**Table 1**  
System Modes of the Hövding 3 with Corresponding Mapping

System Mode	0	1	2	4	5	6	7	8	9
Description	Bluetooth inactive	Bluetooth active	Deployed	SVM Low	SVM Medium	SVM High	Active Misuse	Battery Low	Temperature out of range
Mapping	-	no danger	accident	danger low	danger medium	-	-	no danger	no danger

From this original data several features are derived. The original and derived data is then divided into three groups:

*Group 1* consists of recorded numerical and categorical features. The numerical features are latitude, longitude and altitude which are z-score normalized (Grus, 2019). The derived information from timestamp such

as the weekday, time of day and time of hour in 5 minute intervals treats these values as categorical. *Group 2* consists of the derived features: distance  $d$ , velocity  $v$ , acceleration  $a$ , slope, and a velocity vector (see Table 2). Also some features are derived from timestamps and contain information on which weekday or during which time of day accidents occur (see Table 3).

**Table 2**  
Numerical Feature Calculation

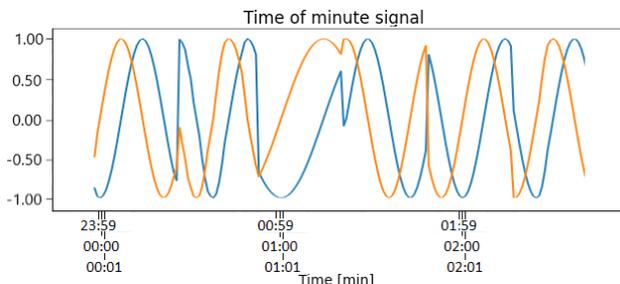
Feature	Calculation	Range	Range after Normalization
distance $d$ [m]	The Haversine formula* (Sinnott, 1984) takes the latitude and longitude of points $P_1 = (\varphi_1, \lambda_1)$ and $P_2 = (\varphi_2, \lambda_2)$ in radians as well as the radius of Earth $r$ as input.	[0; 48.25]	[-2.33; 7.98]
velocity $v$ [m/s]	distance / time difference	[0; 13.55]	[-2.32; 3.74]
acceleration $a$ [m/s <sup>2</sup> ]	distance / time difference <sup>2</sup>	[0; 13.7]	[-1.7; 6.88]
slope	altitude difference / distance	[-0.34; 0.34]	[-5.54; 5.69]
$v_x$ $v_y$	Use angle $\alpha$ determined by three consecutive locations, see Figure 2, and velocity to calculate a velocity vector: $\begin{pmatrix} v_x \\ v_y \end{pmatrix} = v \cdot \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix}$	[-13.4; 10.33] [-10.73; 11.0]	[-3.42; 6.21] [-9.98; 10.31]

$$* d = 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

**Table 3**  
Categorical Features

Feature	Weekday	Time of Day [h]	Time of Hour [5 min]	Day of Year
Extraction from	timestamp	timestamp	timestamp	timestamp
Range	[0; 6]	[0; 23]	[0; 11]	[0; 364]

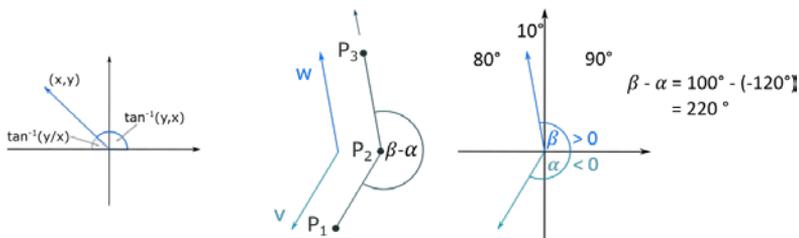
This study models periodicity over time (i.e. the minutes 0, 1, 2 ... 59, 0, 1, 2 of each hour) with sine and cosine transformations, see Figure 2, and treats the periodicity in weeks, time of day and day of year the same.



**Fig. 2.** Modelling of Periodicity of Time of Minute Signal with Sine (Orange) and Cosine (Blue) Transformations

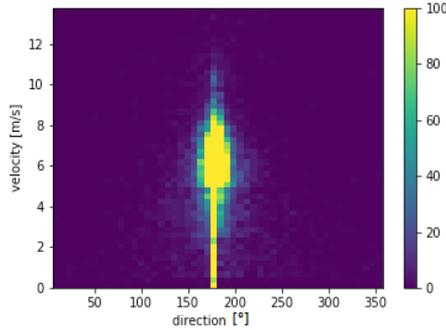
As the range of the amplitudes of the sine signal to the interval  $[-1, 1]$  limits the data, no further normalization has to be done. The Haversine formula (Sinnott, 1984) calculates the distance  $d$  between two latitude/longitude

values which one uses for calculating velocity  $v$ , acceleration  $a$  and slope. This study calculates the angle of three consecutive points in a trip from the cyclist's point of view to capture situations such as turning left or right.



**Fig. 3.** Calculation of the Angle of Three Consecutive Points  $P_1, P_2$  and  $P_3$  in a Trip

Note: Arcos tangent calculates the angle of a vector with respect to the  $x$ -axis, taking into account the position in the quadrants. Three consecutive points  $P_1, P_2$  and  $P_3$  determine the vectors  $\vec{v} = \overline{P_1P_2}$  and  $\vec{w} = \overline{P_2P_3}$ . Tangent calculates the corresponding angles  $\alpha$  and  $\beta$  with respect to the  $x$ -axis. The angle lies between  $0^\circ$  and  $180^\circ$  for positive  $y$ -values and between  $0^\circ$  and  $-180^\circ$  for negative  $y$ -values. Differences of angles of vectors towards the predecessor and successor determine the angle at the location within a tour. The difference of the two angles  $\alpha$  and  $\beta$  determines the angle of the three consecutive points.



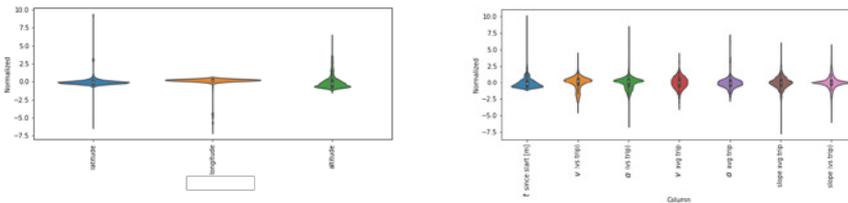
**Fig. 4.**  
Distribution of the Angle of Three Consecutive Points and the Velocity

Straight-ahead driving is common which corresponds to 180°, see Figure 4. Trip segments start with velocity 0 and an angle of 180°.

number telling how long a cyclist is already on the road. This group allows conclusions on changes in the cyclists’ driving behavior per trip and how long a trip might be.

Group 3 consists of average values and differences to these values calculated within the same trip. Also contained in group 3 is a

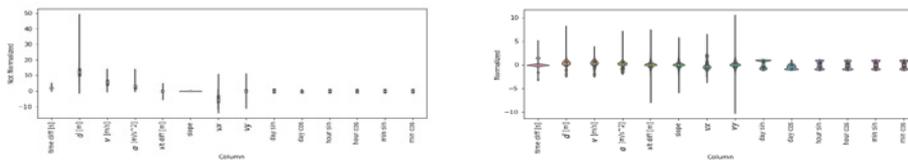
For an overview of the data concentration Seaborn violin plots (Waskom, 2021) display the features of every group:



**Fig. 5.**  
Normalized Numerical Features in Group 1 (Left) and 3 (Right) in a Seaborn Violin Plot  
Source: (Waskom, 2021)

Note: The numerical features in group 1 are latitude, longitude and altitude whereas group 3 consists of time since start  $t$ , velocity  $v$  and acceleration  $a$ , all in comparison to the values in the whole trip as well as average velocity during the trip, average acceleration, slope average and slope in comparison to the overall slope in a trip.

The following two images show the range of the numerical features in group 2 before and after normalization:



**Fig. 6.** Not Normalized (Left) vs. Normalized (Right) Numerical Features in Group 2 using Seaborn Source: (Waskom, 2021)

Note: The numerical features in group 2 are the time difference, distance  $d$ , velocity  $v$ , acceleration  $a$ , altitude difference, slope, velocity in  $x$ - and  $y$ -direction, the weekday, hour and minute with sine and cosine transformations.

### 3.2. Data Cleanup

This study processes the data in several steps; one step is to handle erroneous or missing values (Chollet, 2018). In general this study uses trip segments with a minimum of a 30 second period with about 10 data points.

Within the latitude and longitude values inaccurate values might occur because right after the start of a trip or after loss of connection the recorded position values show non-normal jumps. One method to

determine inaccurate position data is to calculate and filter unrealistic velocity values (see Table 2) and drop their non-natural values.

Inaccurate altitude values can be detected by calculating the slope (see Table 2) and filtering unrealistic values. Known extreme altitude values of the city of Copenhagen are upper and lower limits for altitude values. Inaccurate altitude values are filled by correct altitude values if necessary using the free Open Topo Data API (Nisbet, 2020).



**Fig. 7.** Maps with Location Errors Before and After Cleanup 7a. Single Outlier Error 7b. Drop of Single Outlier and Interpolation of Locations



**Fig. 8.** Maps with Location Errors Before and After Cleanup 8a. Jumps in Locations 8b. Drop of Jumps

The time differences between two consecutive trip locations vary from milliseconds to several minutes, most of them are 2.5 to 3 seconds long. Tiny time

differences result in erroneous velocity calculations; the helmet often records them at the beginning of a trip or after large time gaps. This study drops time differences

smaller than 0.6 seconds in the dataset. Missing recordings can be interpolated considering a limit of 3 seconds and the mean time difference of a trip as interpolation step.

If time differences are larger than 1.8 times the limit, one or more additional points are inserted. Time differences above 20 seconds lead to segmentation of a trip.

**Table 4**

*Effects of the Cleanup Process on Recorded Trip Data*

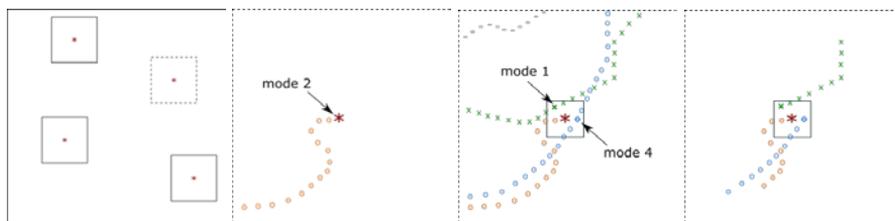
	Before Cleanup	After Cleanup	%
Users	11,223	10,703	95.37 %
Trips	405,221	387,798 trips with 1128043 segments	95.70 %
Locations	103,708,834	100,747,851 (12,518,072 interpolated) 113,265,923 total	97.14 % 109.2 %
Trips with accidents	213	179	84.04 %

### 3.3. Preselection of Trip Segments around Danger Spots for Classification

In classification a category refers to a class and data points refer to samples. In supervised learning the known labels describe the association of a class with a sample (Chollet, 2018). This setup performs a single label multiclass classification where the networks determine the association to one class (Chollet, 2018). This study uses the system modes taken at the end of each trip segment at the nearest point around an accident as labels and so determines whether a certain trip segment ends with a danger spot because the helmet elevates. The four classes each make up 25 %:

- “No danger”: The helmet records “Bluetooth Active”.
- “Danger low”: The helmet records “SVM Low”.
- “Danger medium”: The helmet records “SVM Medium”.
- “Accident”: The helmet records “Elevated”.

For supervised learning every class should have evenly distributed representations. Therefore the number of accidents limits the maximum number of trip segments in each class. 179 accidents remain after the cleanup process which greatly reduces the amount of data. In this study an additional criterion specifies that used trip segments should end in close proximity to an accident. In Figure 9 on the left a rectangle surrounds accident points. The second left picture shows a trip segment which ends with an accident. Within a given square of 8 m around the danger spot, candidates for other trip segment end points occur (see the second right figure). Their segments require a given minimum of predecessors and should only be used once. To receive a balanced data set, one chooses the same numbers of candidates of each class as close as possible. In case a class does not end up with enough trip segment end points, they are selected them around other spots. A maximum number limits the amount of data points in one trip segment (see the right figure).



**Fig. 9.**

Select Trip Segment Data around Danger Spots

Note: For classification the trip segment end points within a given square around danger spots are selected equally distributed among the system modes. The four classes each make up 25 %. The trip segments are restricted by a maximum number of predecessors.

### 3.4. Preselection of Trip Segments around Danger Spots for Binary Classification

For a binary classification into “accident” and “no accident” the multiclass data can be reduced to only two classes. The first one consists of system mode 2 which indicates the helmet is elevated. The second one summarizes system modes which indicate a less dangerous situation. Therefore the first class is called “accident” and the second class is called “no accident”. Three different approaches make up the class “no accident”:

- Approach A chooses an equal amount of nearest trip segment end points around danger spots within all other system modes 1, 4, 5, 8 and 9 (see Table 1).
- Approach B chooses an equal amount of system mode 1 end points around a danger spot.
- Approach C chooses an equal amount of system mode 5 end points around a danger spot.

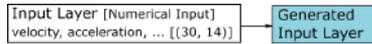
## 4. Neural Networks for Position Data Trips

The input data in our study consists of trip segments each consisting of data of the different feature groups, see section 3.1 to 3.4. One takes the labels at the end of the

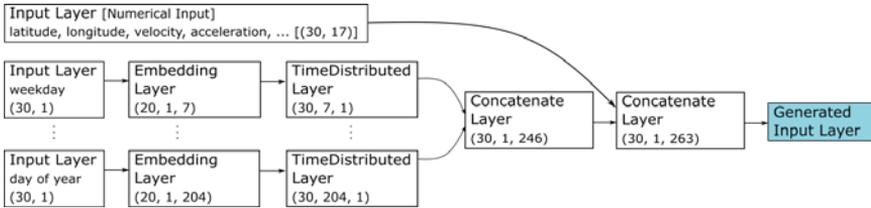
trip segment at the nearest point around an accident. Four classes each make up 25 % of the input data: “no danger”, “low danger”, “medium danger” and “accident” in case of multiclass classification, otherwise only two classes are used. The input data splits itself into a training, a validation and a test set, each containing a balanced number of labels i.e. the system modes at the trip segment end points.

### 4.1. General Setup for all Models

This study tests and compares different models to identify upcoming danger situations for cyclists. It dynamically preprocesses the same dataset for each feature group 1, 2 and 3 and inserts it into different neural network architectures using the same layer as Input Layer for all models. Figure 10 shows an Input Layer for trip segments with 30 data points for feature group 2 with the 14 numerical features displayed in Figure 6. For each categorical feature an Embedding Layer creates a vector with the category size, see Figure 11. A TimeDistributed Layer wraps it in order to allow embedding on a temporal section (TensorFlow, 2021). The concatenated outputs of the numerical input layer and the wrappers provide a predefined output which is used as input for all models.



**Fig. 10.**  
Generated Input Layer for Trip Segments with 30 Data Points, Feature Group 2 and 14 Numerical Features



**Fig. 11.**  
Generated Input Layer for Trip Segments for Feature Groups 1 and 2 with Overall 17 Numerical and 246 Categorical Features

Note: The Input Layers shape consists of (time step × feature). In this example the Input Layer for groups 1 and 2 processes 30 trip segment points, 17 numerical features and 246 categorical features, see Table 3. Each categorical feature is embedded separately, wrapped with a TimeDistributed Layer, afterwards all of them are concatenated. As a result a Concatenate Layer provides the Generated Input Layer which is fed into all models.

Different models use different layers, all of which end with a dense layer with a projection onto the two classes “no danger” and “accident” with a sigmoid activation recommended for binary classification by Chollet (2018). A loss function works as a measurement of the quality of the output when comparing predicted and expected or true values (Chollet, 2018). The BinaryCrossentropy loss function is used in this case. Taking the data and loss function into account an optimizer updates the network’s weights, which in this case is the Adam optimizer (Kingma & Ba, 2014). An early stopping callback function stops the training process to avoid overfitting. For multiclass, single-label classification all models end with a dense layer with a projection onto the four classes with a softmax activation. It uses the categorical crossentropy loss function (Goodfellow, Bengio, & Courville, 2018).

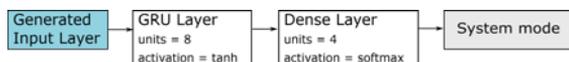
In this study all models are implemented with the Python programming language version 3.9 as well as the Deep Learning library Keras with the TensorFlow version 2.5 backend with CPU support only.

## 4.2. Model Selection

Five different models (see Table A1- A2) categorize whether a certain location is dangerous or not. Every model uses both a single and a chain of dense layers. A Fully Connected Network (FCN) has multiple dense layers whose units connect every other unit in the following layer. Since Dabiri *et al.* (2018) achieves a high accuracy with a CNN in their case, this study also uses a CNN model. Besides computer vision utilizations a CNN can also be fed with time series respectively trip data (Goodfellow *et al.*, 2018).

For processing sequenced data such as a bicycle trip segment, one can also use a recurrent-sequence processing GRU-based model (see Figure 11) as proposed by Cho *et. al* (2014). It achieved the highest accuracy with our data (see Table 6). This type of RNN keeps the information of the current

time step unlike a traditional recurrent unit that drops the information. A GRU-based model is simpler than a Long Short-Term Memory (LSTM) model (Hochreiter & Schmidhuber, 1997). Each stack of layers ends with a dense layer for binary and multiclass, single-label classification.



**Fig. 12.**

*Structure of the GRU-based Model for Multiclass, Single-label Classification*

*Note: The GRU-based model inserts the Generated Input Layer into the GRU Layer with 8 units. Then a Dense layer with a single unit determines the accuracy for binary classification and 4 units for multiclass single-label classification.*

Analysis with an LSTM model show weaker results (see Table A2 in the Appendices) the reason might be the higher complexity of these models compared to the small dataset in this study.

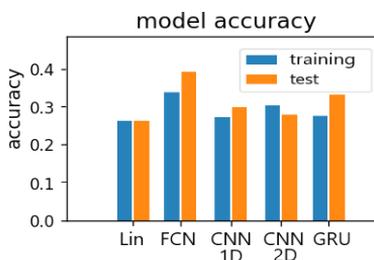
## 5. Results

All of the evaluated models to classify trip segments use different numbers of trip segment points (10, 20 and 30) as input as well as different combinations of feature groups.

### 5.1. Multiclass, Single-label Classification

For multiclass, single-label classification the FCN scored the best result of 39 % accuracy with danger prediction with 10 time steps using feature groups 1, 2 and 3 (see Figure 13). The one-dimensional CNN is referenced as CNN 1D.

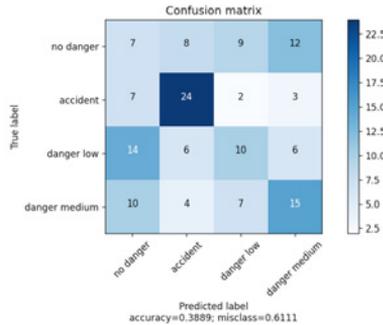
The Table A1 in the Appendices shows the corresponding results with 20 and 30 time steps for each trip segment.



**Fig. 13.**

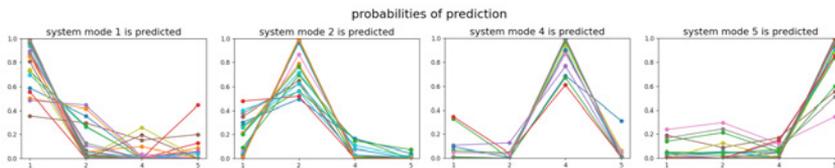
*Accuracy of Danger Prediction for Multiclass, Single-label Classification with Different Models and Feature Groups 1, 2 and 3 for 10 Time Steps*

To further analyze the results of the FCN (see Figure 14). A confusion matrix delivers good results if it has maximum values in the diagonal from the top left to the bottom right.



**Fig. 14.** Confusion Matrix for Multiclass, Single-label Classification with a FCN  
 Note: Most accidents are correctly classified whereas the prediction has difficulties with the categories “no danger”, “danger low” and “danger medium”. The confusion matrix refers to 36 labels per class of the test set.

Figure 14 shows difficulties for classifying “no danger”, “danger low” and “danger medium” but shows a lot of values on the diagonal. But as soon as one looks at the probabilities of the predictions for the individual classes, it can be seen that the results are relatively tight, see Figure 15. Few unique assignments exist.



**Fig. 15.** Probabilities of the Prediction of the Multiclass, Single-label Classification for Each Class

## 5.2. Binary Classification

Using 20 trip segment points binary classification reaches the highest accuracy of 83 % with a GRU-based model with feature

group 2 (see Table 5). With fewer and more time steps the accuracy of the GRU-based model worsens, too (see Table 6). More results of the other models are listed in Table A2 in the Appendices.

**Table 5**  
Results of Binary Classification with 8 Units: **Danger Prediction**

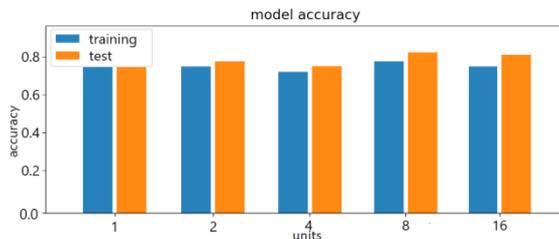
Model	Linear	FCN	CNN 1D	CNN 2D	GRU
Feature Group	2	2	2	2	2
Accuracy – 20 Time Steps	42 %	78 %	79 %	43 %	<b>83 %</b>

The GRU-based model achieves high accuracies with feature group 2 as input for 20 time steps and feature group 1 for 30 time steps for danger prediction (see Table 6). Using feature group 1 results in the lowest accuracy for 10 time steps.

**Table 6**  
Best Results of Binary Classification with 8 Units: **Danger Prediction**

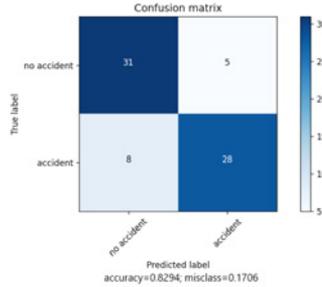
Model	Feature Groups	Accuracy – 10 Time Steps	Accuracy – 20 Time Steps	Accuracy – 30 Time Steps
GRU	1	67 %	75 %	82 %
	2	75 %	<b>83 %</b>	78 %
	3	76 %	72 %	71 %
	1, 2	78 %	79 %	78 %
	1, 3	71 %	76 %	78 %
	2, 3	78 %	78 %	76 %
	1, 2, 3	75 %	78 %	78 %

Tests for the GRU-based model with different units reveal that the results are similar.



**Fig. 16.**  
Similar Results of GRU-based Network with Different Units

A confusion matrix shows the results of the GRU-based model for binary classification:



**Fig. 17.**

*Confusion Matrix for Binary Classification with a GRU-based Model*

Note: The results of the confusion matrix are good: Most values are in the diagonal from top left to bottom right. 31 spots with no accident are classified correctly as well as 28 accident spots. 8 accident spots are falsely classified and 5 spots with no accident is classified as an accident. The confusion matrix refers to 36 labels per class of the test set.

The confusion matrix in Figure 17 shows most of the values along the diagonal. This allows the conclusion that a prediction of the system mode can be made based on the trip time series data, which is influenced by the driving behaviour. There is an exception of 8 cases where no danger is predicted but an accident still occurs which can be seen in the first column in the second row. These accidents might

have occurred due to external influences or human failure. Figure 18 shows one of these accident locations located on a bridge next to a street sign. In the top view on the left the shadow of the street sign is shown. As the location is in close proximity to the obstacle, it might have caused the accident. Pedestrians could have also been involved. The exact cause of the accident cannot be derived from the data.



**Fig. 18.**

*Google Earth and Street View of Latitude/Longitude Values (55.686909, 12.563077)*

Source: (Google, 2022b, 2022d)

Another accident occurred next to a curb with cobblestones in close proximity to a pedestrian crossing, see Figure 19. The curb, cobblestones and mobile display

are obstacles on which a cyclist might get stuck. Turning is also a risk factor as well as the pedestrians at locations like this one.



**Fig. 19.**

*Google Earth and Street View of Latitude/Longitude Values (55.680494, 12.587165)*

*Source: (Google, 2022a, 2022c)*

## 6. Conclusion

Multiclass single-label classification danger prediction is done using a linear regression approach, a single or a chain of dense layers, a CNN, a GRU or a LSTM. The best result to classify the four classes “no danger”, “danger low”, “danger medium” and “accident” is 39 % accuracy which the FCN model achieves using all the features as input. For the binary classification the same models are used as for multiclass, single-label classification. The GRU-based model achieves a high accuracy of 83 % using the calculated features as input. A more complex model such as a CNN or LSTM offers no added value. When adding feature group 2 to the input features, the most accuracy can be gained. Future studies could leave out feature group 1 as input for neural networks.

The results in this paper show that a prediction of the system mode and thus a dangerous situation can be made based on the trip data. This thesis cannot be applied where the neural network predicts no accident, but an accident happens. By analysis of these two accident locations with Google Earth and Street View the study shows that cobblestones, curbs, pedestrians and street signs may be external influences on the accidents. The binary classification supports the thesis.

The present research is limited by the amount of accidents recorded with the Hövding 3 which naturally limits the amount of input data for the neural networks. User specific data could be taken into account giving the respective user information about general driving behavior and whether or not a dangerous situation emerges from a deviation from that behavior. This data would not only set the cyclists’ behavior in the same trip in comparison, but the behavior in the cyclists’ all previous trips.

Around the danger spots there are few system mode 6 (“Danger High”) occurrences and therefore those trip segments are ignored. When more data is collected, they are included. In future studies highly dangerous spots are analyzed instead of accident spots.

The prediction algorithm can be integrated into a real time warning system for cyclists. A warning would appear in case a cyclist shows unusual and dangerous driving behavior. The cyclists could also be informed about upcoming construction sites and other obstacles. The results of this work might be used to report danger spots to authorities which might be interested in making them safer.

## Acknowledgements

This study collaborates with the company Høvdning Sverige AB which provided us with the data collected by the helmet Høvdning 3.

## Appendices

**Table A1**

Results of Multiclass, Single-label Classification with 8 Units: **Danger Prediction**

Model	Feature Groups	Accuracy – 10 Time Steps	Accuracy – 20 Time Steps	Accuracy – 30 Time Steps
Linear	1, 2, 3	27 %	24 %	35 %
FCN	1, 2, 3	<b>39 %</b>	37 %	37 %
CNN 1D	1, 2, 3	33 %	37 %	37 %
CNN 2D	1, 2, 3	29 %	32 %	29 %
GRU	1, 2, 3	34 %	34 %	<b>38 %</b>
LSTM	1, 2, 3	37 %	34 %	38 %

**Table A2**

Results of Binary Classification with 8 Units

Model	Feature Group	Accuracy – 10 Time Steps	Accuracy – 20 Time Steps	Accuracy – 30 Time Steps
Linear	2	55 %	42 %	43 %
FCN	2	79 %	78 %	<b>69 %</b>
CNN1D	2	78 %	76 %	79 %
CNN2D	2	53 %	46 %	43 %
GRU	2	75 %	<b>83 %</b>	78 %
LSTM	2	74 %	76 %	74 %

**Table A3**

Results of Binary Classification Danger Prediction with 8 Units

Model	Time Steps	Feature Groups	Accuracy – Scenario A	Accuracy – Scenario B	Accuracy – Scenario C
FCN	20	2	66 %	63 %	76 %
	30	2	65 %	69 %	79 %
GRU	20	2	71%	69 %	<b>83 %</b>
	30	2	74 %	68 %	78 %

## References

- Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. Available from Internet: <<https://arxiv.org/pdf/1409.1259>>.
- Cycling Embassy of Denmark. 2018. The Bicycle Account 2018 Copenhagen City of Cyclists. Available from internet: <<https://cyclingsolutions.info/wp-content/uploads//2020/12/CPH-Bicycle-Account-2018.pdf>>.
- Chollet, F. 2018. *Deep Learning with Python*. Manning Publications Co. USA. 361 p.
- Dabiri, S.; Heaslip, K. 2018. Inferring transportation modes from GPS trajectories using a convolutional

- neural network, *Transportation Research Part C: Emerging Technologies* 86: 360–371.
- Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *KDD* 96(34): 226-231.
- Goodfellow, I.; Bengio, Y.; Courville, A. 2016. *Deep Learning*. MIT Press. USA. 800 p.
- Google. 2022a. Google Maps/Google Earth (55.680494. 12.587165). Available from Internet: <<https://www.google.de/maps/place/55%C2%B040'49.8%22N+12%C2%B035'13.8%22E/@55.6806489,12.5869092,53m/data=!3m1!1e3!4m5!3m4!1s0x0:0xd3fe721709c52fa-6!8m2!3d55.680494!4d12.587165>>.
- Google. 2022b. Google Maps/Google Earth (55.686909, 12.563077). Available from Internet: <<https://www.google.de/maps/place/55%C2%B041'12.9%22N+12%C2%B033'47.1%22E/@55.6869197,12.5628097,53m/data=!3m1!1e3!4m3!1m7!3m6!1s0x0:0xae13b3b7bd52ac1a-!2zNTXCsDQxjzEyLjkiTiAxMsKwMzMnNDcuMSJF!-3b1!8m2!3d55.686909!4d12.563077!3m4!1s0x0:0xae1-3b3b7bd52ac1a!8m2!3d55.686909!4d12.563077>>.
- Google. 2022c. Google Street View (55.680494. 12.5-87165). Available from Internet: <[https://www.google.de/maps/@55.6805485,12.5871428,3a,75y,158.7h,-76.44t/data=!3m6!1e1!3m4!1sPCivWZTPWpm\\_D39C2vGkGA!2e0!7i16384!8i8192](https://www.google.de/maps/@55.6805485,12.5871428,3a,75y,158.7h,-76.44t/data=!3m6!1e1!3m4!1sPCivWZTPWpm_D39C2vGkGA!2e0!7i16384!8i8192)>.
- Google. 2022d. Google Street View (55.686909, 12.563077). Available from Internet: <<https://www.google.de/maps/@55.6869848,12.5630764,3a,75y,-173.75h,80.98t/data=!3m6!1e1!3m4!1sTNadfLNzcX- unbSEQUjRuXw!2e0!7i16384!8i8192>>.
- Grus, J. 2019. *Data science from Scratch*. O'Reilly Media, Inc. USA. 376 p.
- Hochreiter, S.; Schmidhuber, J. 1997. Long short-term memory, *Neural Computation* 9(8): 1735–1780.
- Holmgren, J.; Knapen, L.; Olsson, V.; Masud, A. P. 2020. On the use of clustering analysis for identification of unsafe places in an urban traffic network, *Procedia Computer Science* 170: 187-194.
- International Transport Forum. 2013. *Cycling, Health and Safety*. Organisation for Economic Co-operation and Development Publishing. France. 248 p.
- Jiang, X.; Souza, E. N. d.; Pesaranghader, A.; Hu, B.; Silver, D. L.; Matwin, S. 2017. TrajectoryNet: An Embedded GPS Trajectory Representation for Point-based Classification Using Recurrent Neural Networks. Available from Internet: <<https://arxiv.org/pdf/1705.02636v2>>.
- Karakaya, A. S.; Hasenburg, J.; Bermbach, D. 2020. SimRa: Using crowdsourcing to identify near miss hotspots in bicycle traffic, *Pervasive and Mobile Computing* 67: 101-197.
- Kingma, D. P.; Ba, J. 2014. Adam: A Method for Stochastic Optimization. Available from Internet: <<https://arxiv.org/pdf/1412.6980>>.
- Krizhevsky, A.; Sutskever, I.; Hinton, G. E. 2017. Imagenet classification with deep convolutional neural networks, *Communications of the ACM* 60(6): 84–90.
- Lindqvist, S.; Roos, J. 2020. Identification of areas with increased risk of accidents for cyclists based on bicycle helmet data [In Norwegian: Identifiering av områden med förhöjd olycksrisk för cyklistar baserad på cykelhjälmdata]. Available from Internet: <<https://www.diva-portal.org/smash/get/diva2:1480296/FULLTEXT01.pdf>>.
- MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 281–297.
- Nisbet, A. 2020. Open Topo Data. Available from Internet: <<https://www.opentopodata.org/#public-api>>.

- Saiprasert, C.; Pholprasit, T.; Thajchayapong, S. 2017. Detection of Driving Events using Sensory Data on Smartphone, *International Journal of Intelligent Transportation Systems Research* 15(1): 17–28.
- Sinnott, R. W. 1984. Virtues of the Haversine, *Sky and Telescope* 68(2): 158.
- Steinwart, I.; Christmann, A. 2008. *Support Vector Machines*. Springer. USA. 601 p.
- STRADA. 2021. About the Strada accident database [In Swedish: Om olycksdatabasen Strada]. Available from Internet: <<https://www.transportstyrelsen.se/STRADA>>.
- Sun, S.; Chen, J.; Sun, J. 2019. Traffic congestion prediction based on GPS trajectory data, *International Journal of Distributed Sensor Networks* 15(5): 155014771984744.
- TensorFlow. 2021a. Time series forecasting; TensorFlow Core. Available from Internet: <[https://www.tensorflow.org/tutorials/structured\\_data/time\\_series?hl=en](https://www.tensorflow.org/tutorials/structured_data/time_series?hl=en)>.
- TensorFlow. 2021b. Module: tf.keras; TensorFlow Core v2.7.0. Available from Internet: <[https://www.tensorflow.org/api\\_docs/python/tf/keras](https://www.tensorflow.org/api_docs/python/tf/keras)>.
- Wang, Y.; Qin, K.; Chen, Y.; Zhao, P. 2018. Detecting Anomalous Trajectories and Behavior Patterns Using Hierarchical Clustering from Taxi GPS Data, *ISPRS International Journal of Geo-Information* 7(1): 25.
- Waskom, M. L. 2021. Seaborn: statistical data visualization, *Journal of Open Source Software* 6(60): 3021.
- Watson, A.; Watson, B.; Vallmuur, K. 2015. Estimating under-reporting of road crash injuries to police using multiple linked data collections, *Accident; Analysis and Prevention* 83: 18–25.