

Gathering Bicycle Ride Data and Using it for the Good

vorgelegt von

M.Sc.

Ahmet-Serdar Karakaya

ORCID: 0000-0002-2040-2782

an der Fakultät IV – Elektrotechnik und Informatik

der Technischen Universität Berlin

zur Erlangung des akademischen Grades

Fakultät IV

Technische Universität Berlin

Promotionsausschuss:

Vorsitzender: Prof. Dr. Stefan Tai

Gutachter: Prof. Dr. David Bermbach

Gutachter: Prof. Dr. Heather Kaths

Gutachter: Prof. Dr. Odej Kao

2

Tag der Wissenschaftlicher Aussprache: 21.03.2024

März 2024

I

Sworn Affidavit

I hereby declare that the thesis submitted is my own, unaided work, completed without any unpermitted external help. Only the sources and resources listed were used.

The independent and unaided completion of the thesis is affirmed by affidavit:

Berlin, January 1st, 1970

Jana Eggers

Abstract

An increased modal share of bicycle traffic is a key mechanism to reduce emissions and solve traffic-related problems. However, a lack of (perceived) safety keeps people from using their bikes more frequently. To improve safety in bicycle traffic, city planners need an overview of accidents, near miss incidents, and bike routes. Such information, however, is currently not available. In this paper, we describe SimRa, a platform for collecting data on bicycle routes and near miss incidents using smartphone-based crowdsourcing. We also describe how we identify dangerous near miss hotspots based on the collected data and propose a scoring model.

Kurzdarstellung

Eine Erhöhung des Fahrradverkehrs ist ein wichtiges Instrument zur Verringerung der Emissionen und zur Lösung verkehrsbedingter Probleme. Allerdings hält ein Mangel an (gefühlter) Sicherheit die Menschen davon ab, ihr Fahrrad häufiger zu benutzen. Um die Sicherheit im Radverkehr zu verbessern, benötigen Stadtplaner einen Überblick über Unfälle, Gefahrensituationen und Radwege. Solche Informationen sind jedoch derzeit nicht verfügbar. In diesem Beitrag beschreiben wir SimRa, eine Plattform zur Sammlung von Daten über Fahrradrouten und Gefahrensituationen mithilfe von Smartphone-basiertem Crowdsourcing. Außerdem beschreiben wir, wie wir auf der Grundlage der gesammelten Daten gefährliche Hotspots für Gefahrensituationen identifizieren und ein Scoring-Modell vorschlagen.

Table of Contents

Declaration	I
Abstract	III
Zusammenfassung	IV
List of Figures	IX
List of Tables	XVI
List of Listings	1
I Foundations	3
1 Introduction	5
1.1 Problem Statement	6
	V

1.2	Contributions of this Thesis	7
1.3	Structure of this Thesis	7
2	Background (rename?)	9
2.1	Crowdsensing and Citizen Science	9
2.2	Deep Learning	10
2.3	SUMO	11
2.4	Surface Quality	12
2.5	Bicycle Traffic Safety	13
3	Contributions	17
3.1	CycleSense	17
3.1.1	Detecting Near Miss Incidents	20
3.1.2	Evaluation of CycleSense	28
3.1.3	Discussion of CycleSense	34
3.1.4	Related Work of CycleSense	40
3.1.5	Summary of CycleSense	41

3.2 CycleQuality	42
3.2.1 Data Processing Pipeline	45
3.2.2 Using Road Surface Quality Information	48
3.2.3 Evaluation of CycleQuality	52
3.2.4 Discussion of CycleQuality	60
3.2.5 Related Work of CycleQuality	63
3.2.6 Summary of CycleQuality	65
3.3 Cyclist Model for SUMO	66
3.3.1 Cycling Behavior in SimRa and SUMO	69
3.3.2 Improving SUMO's Bicycle Simulation	76
3.3.3 Evaluation of the Cyclist Model for SUMO	79
3.3.4 Discussion of the Cyclist Model for SUMO	88
3.3.5 Generalizability	91
3.3.6 Related Work of Bicycle Simulation for SUMO	92
3.3.7 Summary of Bicycle Simulation for SUMO	95

4 Conclusions	113
4.1 Discussion	113
4.2 Summary and Outlook	113
A Technologies and Concepts of a Larger Context	115
A.1 Finding Consensus in a Distributed System	115
B Acronyms	117
C Lexicon	121
D Listings	123
D.1 Configuration for Node A	123
Bibliography	125

List of Figures

3.1	The accelerometer readings of an example ride. The encircled spots could indicate an incident but also driving over a curb.	21
3.2	The model architecture in a nutshell: Using subnetworks for feature extraction of the different sensors (accelerometer, gyroscope, and Global Positioning System (GPS)) and afterwards a fusion network to combine the features for final incident recognition [21].	25
3.3	Results of the hyperparameter optimization for the four variables f (the window size), the number of Recurrent Neural Network (RNN) units used, the RNN cell type utilized, and the learning rate from left to right.	26

3.4 Comparison of the baselines and common model architectures used in the context of Deep Learning (DL) for Time Series Classification (TSC) [46] with the CycleSense model. Note that all of these models have been trained and evaluated on rides contained in the SimRa data set that have been recorded with more recent versions of the SimRa Android app.	32
3.5 Comparison of CycleSense trained only on the newer Android rides or individual CycleSense models trained for each part of the data set. Note that only the newer Android data set was manually cleaned.	34
3.6 Comparison of the performance of CycleSense trained on Berlin and trained on the other regions combined.	35
3.7 Impact of different preprocessing and training steps on the performance of CycleSense.	36
3.8 Box plot showing the empirical measurement times observed in the SimRa data set for rides recorded with different versions of the SimRa app.	38
3.9 Visualization of the sensor data of a simulated heavy braking incident vs. a moderate braking event before and after the moving average has been applied.	39
3.10 Overview of our surface quality analysis pipeline.	45

3.11 With a slider in the settings menu, the weight of the surface quality in the routing can be set.	49
3.12 A visualization of the output file showing the surface quality of the boxes when hovering over them with the mouse cursor. Color coding is based on the average value.	51
3.13 The Probability Density Function of the Derived Surface Qualities shows that these segments have very undisputed road surface quality values, since they are either very good or very bad.	53
3.14 The Probability Density Function of the Derived Surface Qualities shows that these segments have confusing road surface quality values: Depending on the ride, they appear to have either very good or very bad surface quality (multiple peaks).	55
3.15 A bus lane, a bus station and a bicycle lane on the sidewalk can create different results in a small area. (Source: Apple Maps)	57
3.16 The Probability Density Function of the Derived Surface Qualities shows that these segments have very smooth road surfaces, although they are paved with cobblestones.	59

3.17 Straßburger Straße and Goethestraße in Berlin are paved with large cobblestones and the roads are contested by cars that are parking or searching for a parking spot. In contrast, the sidewalks are paved with flat-surfaced paving stones and are quiet due to the absence of shops, restaurants or cafes. (Source: Apple Maps)	97
3.18 Histogram of the empirical average velocity capabilities of cyclists found in the SimRa dataset. The average velocity of all bicycle rides after the preprocessing is 4.38 m/s with a standard deviation of 0.89 and a median of 4.42 m/s.	98
3.19 Histogram of maximum acceleration capabilities found in the empirical SimRa dataset and their respective distribution functions. The red scalar represents the default value in SUMO.	99
3.20 Histogram of maximum deceleration capabilities found in the empirical SimRa dataset and their respective distribution functions. The red scalar represents the default value in SUMO.	100
3.21 Histogram of maximum velocity capabilities found in the empirical SimRa dataset and their respective distribution functions. The red scalar represents the default value in SUMO.	101

3.22 Qualitative comparison between the SUMO default intersection model and real world data given by SimRa for the intersection between Alexanderstraße and Karl-Marx-Allee in Berlin. SimRa shows two distinct left-turn paths (i.e., a direct and an indirect one) whereas SUMO default only models the direct path.	102
3.23 Excerpt from the <i>Mehringdamm</i> scenario in SUMO. The scenario was created using OSM data only.	103
3.24 Histogram of SUMO's, SimRa's, and our approach's observed maximum accelerations inside the example <i>Frauentorgraben</i> scenario. While the maximum accelerations are heterogeneously distributed in the real-world data and our approach, the default values are clustered. Here, the observed accelerations deviate from the configured default value (1.2 m/s^2) due to traffic effects inside the simulation.	104
3.25 Histogram of SUMO's, SimRa's, and our approach's observed maximum decelerations inside the example <i>Dachauer Straße</i> scenario. While the maximum decelerations are heterogeneously distributed in the real-world data and our approach, the default values are clustered. Here, the observed decelerations deviate from the configured default value (3m/s^2) due to traffic effects inside the simulation.	105

- 3.26 Histogram of SUMO’s, SimRa’s, and our approach’s observed maximum velocities inside the example *Oranienstraße* scenario. While the maximum velocities are heterogeneously distributed in the real-world data and our approach, the default values are clustered. 106
- 3.27 Qualitative comparison between the results of our approach for the intersection model and real-world raw GPS data given by SimRa for the intersection between Alexanderstraße and Karl-Marx-Allee in Berlin. SimRa shows two distinct left-turn paths (i.e., a direct and an indirect one), which are also modeled by our models. Also the three different groups have different shares between direct and indirect left-turns. . 107
- 3.28 ECDFs of the measured durations for crossing the scenario *Warschauer Straße*. It is apparent that our models for all, medium (*m*) and fast (*f*) cyclists outperform SUMO’s default as the measured durations converge towards the real-world data. Only our fast cyclist model (*f*) is just marginally better than SUMO’s default. 108
- 3.29 ECDFs of the measured durations for crossing the scenario *Mehringdamm*. The results when using our models for all, medium (*m*) and fast (*f*) cyclists are only slightly more realistic than when using the standard SUMO model. However, when the *direct* path is blocked for cyclists, the simulation results outperform the default approach. . . 109

3.30 ECDFs of the measured durations for crossing the scenario <i>Alexanderstraße</i> . Here, our new approach is not more precise than SUMO default behavior.	110
3.31 Intersection Mehringdamm/Gneisenaustraße: a traffic island obstructs the direct turn path (dashed line) and, thus, makes the indirect path (solid line) more likely to be used.	111

List of Tables

3.1	Comparison of the baselines and common model architectures used in the context of DL for TSC [46] with the CycleSense model. See the discussion in Section ?? about the usefulness of various metrics. Note also that all of these models have been trained and evaluated on rides contained in the SimRa data set that have been recorded with more recent versions of the SimRa Android app. For all models the threshold which optimizes Youden's index[98] were chosen.	33
3.2	Surface Quality Analysis Evaluation Results Showing Mean, Median and Standard Deviation of Sections With Clear Results	54
3.3	Surface Quality Analysis Evaluation Results Showing Mean, Median and Standard Deviation of Sections With Mixed Results	56
3.4	Surface Quality Analysis Evaluation Results Showing Mean, Median and Standard Deviation of Sections With (Seemingly) Confusing Results	60

3.5	Most important attributes of entities of the SimRa dataset	70
3.6	Most important attributes of our simulation scenarios.	81
3.7	Results Overview Maximum Acceleration Comparing SUMO Default, Our Approach (All), Our Approach (Slow), SimRa (All) and SimRa (Slow)	83
3.8	Results Overview Maximum Deceleration Comparing SUMO Default, Our Approach (All), Our Approach (Medium), SimRa (All) and SimRa (Medium)	84
3.9	Results Overview Maximum Velocity Comparing SUMO Default, Our Approach (All), Our Approach (Fast), SimRa (All) and SimRa (Fast)	86

List of Listings

D.1 Configuration for Node A	123
--	-----

Part I

Foundations

Chapter 1

Introduction

The global climate crisis is and will be one of mankind's main challenges in the coming decades. More and more societies worldwide are aware of this problem and search for solutions to adapt to the new changes that comes with the climate change and try to decrease the impact of the global climate crisis. For the latter, the main approach is to reduce greenhouse gas emissions such as CO_2 and NO_x . This is why cities worldwide try to increase the modal share of bicycle traffic and by doing so, decrease the usage of individual motorized transport, which is one of the main greenhouse gas emitters in urban areas (CITATION NEEDED). This, however, puts cyclists, city planners and politicians into a difficult spot. Since cities were developed with the individual motorized transport as the main transportation mode in mind, the traffic infrastructure highly favors the usage of cars. Because of that, cyclists often

have a hard time commuting, since they have to cycle on roads that were made for cars, which often puts them into dangerous and stressful situations. Although crash statistics do not convey an increased danger for cyclists (CITATION NEEDED), studies have shown, that the perceived safety of commuting with a bicycle is very low (CITATION NEEDED) and one of the main reasons why people prefer other transportation modes to cycling (CITATION NEEDED). It is up to city planners to improve the traffic infrastructure for cyclists, but it is not easy to pinpoint the most problematic areas, that need to be prioritized for them to be fixed, which is needed because of the limited resources available to the city planning department. Besides, city planners are bound by city planning codes and rules, which oftentimes also reflect the car-centric approach. This passes the responsibility to the politicians, who need to change the city planning codes and rules and also increase the funds for bicycle infrastructure. For this, however, they need more than anecdotal evidence, that the perceived safety in bicycle traffic is low, so that they can change the legislation and increase funding in favor of bicycle infrastructure.

1.1 Problem Statement

To enable city planners and politicians to do evidence-based decision-making, they need a) to be aware of the problems in bicycle traffic and b) to know where and how to improve the bicycle infrastructure. There are surveys that show that the

perceived safety in bicycle traffic is low (MULTIPLE CITATIONS) DO I NEED TO GIVE EXAMPLES HERE FOR SURVEYS?

1.2 Contributions of this Thesis

1.3 Structure of this Thesis

Chapter 2

Background (rename?)

2.1 Crowdsensing and Citizen Science

Other edge-assisted data analysis work includes Mei et al. [**mei2017ultraviolet**], who measure UV radiation based on smartphone cameras and crowdsensing, Cao et al. [**cao2015fast**] who use motion sensors to detect strokes in patients falling to the ground, and Pham et al. [**pham2015a**] implement a smart parking system by equipping parking spots with RFID chips to track their occupancy state. There are also multiple publications on placing different components of an edge-to-cloud data processing pipeline in various use cases, e.g., [**paper·lujic·intrarasafed5g**, **paper·pfandzelter·zero2fog**, **hattab2019optimized**].

2.2 Deep Learning

One of the most common tasks for [TSC](#) are Natural Language Processing, speech recognition, and audio recognition in general. Another field that deals with this problem is Human Activity Recognition ([HAR](#)) which is concerned with identifying the specific activity of a human based on sensory time series data. Time windows of a few seconds are classified into activity categories (e.g., walking, sitting, running, lying). While various feature extraction and pattern recognition methods have been successfully applied in the past in this context [17], those approaches have constraints such as hand-crafted feature extraction, being able to only learn shallow features [96], or the requirement for large amounts of well-labeled data for model training [91]. Deep Learning techniques and more specifically Convolutional Neural Networks ([CNNs](#)) have recently proven to overcome these issues and deliver convincing results in the context of [HAR](#) [[Ronao2015](#), 91]. Also, implementations of [RNNs](#) such as Long Short-Term Memory ([LSTM](#)) [[Tao2016](#), 97] have proven to be successful. Yao et al. [97] use a combination of [RNNs](#) and [CNNs](#) on top of a Sensor-based Fusion ([SF](#)) approach after preprocessing their input data using a Discrete Fourier Transform ([DFT](#)).

2.3 SUMO

SUMO is an open source traffic simulation tool that offers macroscopic as well as microscopic simulation of vehicle mobility [60]. SUMO includes models for different types of “vehicles”, including, among others, cars, bicycles, and even pedestrians. Due to its large feature set, it has become the de-facto standard for traffic simulation and is used even beyond the transport community, e.g., [9].

Traffic scenarios are, among other things, defined by road networks and vehicle traffic. The road network includes roads and their (sub-)lanes as well as exclusive lanes for cyclists and pedestrians, or road-side infrastructure such as traffic lights. Furthermore, connections between these lanes and traffic lights can be configured.

When modeling vehicle traffic, users specify demand for a specific road segment per vehicle type and can adjust vehicle-specific parameters of SUMO’s simulation model to control their respective behavior. In general, vehicle parameters are usually specified in the vehicle type declaration (*vType*), applying the changes to all instances of the respective *vType*, e.g., to all cars. An alternative, however, is to obtain multiple *vType* realizations which typically differ in at least one parameter by using so-called *vTypeDistributions*. This way, when spawning a new vehicle, SUMO randomly picks a specific *vType* from the *vTypeDistribution* and instantiates the vehicle’s parameters accordingly, e.g., cars can thus have individual maximum velocities.

In SUMO, vehicle behavior is, among other things, defined by *Car Following (CF) models* for the longitudinal kinematic behavior, *Lane Change (LC) models* for the lateral kinematic behaviour, and *junction models* for the behavior at junctions and intersections.

Despite including several of these models for cars and trucks, SUMO does not provide a dedicated movement model for cyclists. Instead, cyclists are simulated by modeling them either as slow cars or fast pedestrians. Both of these approaches use movement models of the corresponding vehicle type and adapt their respective shape and kinematic characteristics (e.g., velocity and acceleration profiles) to match cyclists. While this is obviously a rough approximation, it is unlikely to reflect the behavior of real-world cyclists [37].

2.4 Surface Quality

Traffic departments frequently analyze the road surface quality to find out road segments that need to be repaired to increase the traffic safety. There are mainly three types of methods used for, which we briefly introduce here. *Profilographs* are one of the oldest devices used to measure road surface quality. They consist of a profiling wheel in the center and multiple support wheels held together by a rigid frame. They measure the road quality by tracking the vertical movement of the profiling wheel. The higher the vertical movement of that wheel, the worse the

surface quality of the road. This method of measuring surface quality is very slow, since the profilograph has to be pulled very slowly by another vehicle. *Scanner-based systems* are more sophisticated and rely either on accelerometers measuring the vibrations caused by driving on a specific surface, lasers scanning the surface in front or beneath the vehicle, cameras making photos later to be analyzed with the help of computer vision and photogrammetry, or a combination of the aforementioned methods. This system provides the highest resolution and accuracy, however it is also the most costly and complex one. More recently, *smartphone-based systems* are gaining traction due to their cost-efficiency. A smartphone is being attached inside the car and the vibrations caused by the road surface are recorded. This is the least accurate system, since the measured vibrations are very indirect, due to car tires and suspension.

However, these systems are impractical for measuring bicycle road surface quality, because cars are not suited to drive on bicycle roads and these systems are too costly. Hence, we propose a novel system using crowdsourced smartphone bicycle ride data.

2.5 Bicycle Traffic Safety

Studies on safety in bicycle traffic often rely on crowdsourcing, e.g., using Strava¹ as a data source. While the Strava data are heavily biased towards recreational trips, some studies rely on them for analyzing various aspects of safety in bicycle traf-

¹<https://www.strava.com/>

fic, e.g., [**Hochmair2019**, **Ferster2021strava**]. Blanc and Figliozi [14, 15] study the perceived comfort levels of cyclists based on routes taken. Wu et al. [94] predict perceived bicycle safety by combining data from, e.g., OpenStreetMap², crime statistics, and parking volumes. Similarly, Yasmin and Eluru [**Yasmin2016**] use various open data sources to characterize and provide estimates for bicycle safety in urban areas. He et al. [**He2018**] analyze trajectories of a bike sharing service to detect events of illegally parked vehicles which frequently affect cycling safety. Figliozi et al. [**figliozi2019evaluation**] evaluate video recordings to identify safety and delay related problems in bicycle and bus traffic. Unsurprisingly, bicycles crossing bus lanes can cause slight delays for the buses.

In addition, Kobana et al. [**kobana2014detection**] focus on the detection of road damage via smartphone data, and Candefjord et al. [**candefjord2014using**] evaluate the development of a crash detection algorithm for cycling accidents.

Most closely related to our work, Aldred and Goodman [3] analyze near-miss incidents using road diaries of cyclists and our previous work [50] proposed the SimRa platform as well as the heuristic for detecting (near-miss) incidents. A first extension of our own detection approach [75] developed an Artificial Neural Network (**ANN**) to solve the problem of incident detection on the SimRa data set. While offering some improvement over the original heuristic, CycleSense clearly outperforms that approach. Furthermore, Ibrahim et al. [**ibrahim2021cycling**] discuss the potential

²<https://www.openstreetmap.com/>

of detecting incidents using image and video data in combination with computer vision techniques.

Chapter 3

Contributions

3.1 CycleSense

In recent years, more and more cities worldwide aim to reduce the modal share of car traffic in favor of bicycle traffic to reduce NO_x, CO₂, and particulate matter emissions, to reduce traffic jams, and to free up space that is urgently needed for other purposes ranging from vegetation that provides natural cooling in a heating world to new flats for a growing population.

A key mechanism to support this is to make bicycle traffic more attractive. In practice, however, what keeps people from using their bikes more frequently is a lack of safety or perceived safety in cities with a car-centric traffic infrastructure [3].

Hence, city planners urgently require an overview of safety and perceived safety in their city.

Aside from actual accidents, an often overlooked aspect of cycling safety are near miss incidents¹ such as close passes or near dooring [3, 50]. Information on these have previously not been available or have not been easily accessible as they are distributed over private video recordings, social media posts, public CCTV footage, and other sources. In 2019, we therefore launched the SimRa² project in which cyclists record their rides and annotate them with incident information via a smartphone app [50].

While our previous approach [50] already used a rudimentary heuristic for automatically detecting incidents based on acceleration sensors, it is inherently limited resulting in significant manual annotation efforts which makes it unattractive for some groups of cyclists – in particular, elderly cyclists who have problems using smartphone apps and cyclists with significant daily mileage for whom the labeling approach is too much effort. To increase participation, it is hence crucial to decrease manual efforts through an improved pre-detection of incidents.

As a first step towards this goal, we supervised a master’s thesis which developed a neural network-based method for incident detection using the public SimRa data set³ [75] which, however, does not show the desired detection quality despite being an improvement over our original heuristic. Both detection methods are currently

¹In the remainder of this paper, we will also refer to them as “incidents”.

²SimRa is a German acronym for safety in bicycle traffic.

³<https://github.com/simra-project/dataset>

used in the live version of the app, hence, we will later use them as baseline in our evaluation. Besides that, there are alternative approaches for quantifying the perceived safety of bicycle traffic, e.g., [14, 15, 94], but to the best of our knowledge no other methods are based on mobile motion sensory data.

Since its start in 2019, the SimRa data set has grown to more than 65,000 rides with almost 30,000 reported incidents. These amounts of data now enable the application of much more sophisticated methods, namely Machine Learning ([ML](#)) and/or [DL](#), for automatic incident detection. Therefore, we here propose CycleSense – a [DL](#) model trained on smartphone sensory data from the SimRa data set to detect incidents in the SimRa app. We hope, that it will lead to more reported incidents due to an easier reporting, and make the following contributions:

- We propose an approach that combines signal processing and [ML](#) techniques to detect incidents based on motion sensor data of cyclists with an Area under the curve ([AUC](#)) Receiver Operating Characteristic ([ROC](#)) of 0.906 (Section ??).
- We evaluate our approach using the SimRa data set and compare it to two baselines as well as common Deep Learning models used in the context of Time Series Classification (Section ??).
- We discuss to which degree our approach can automate incident detection and which additional sensors are needed for full automation (Section ??).

3.1.1 Detecting Near Miss Incidents

This section describes the process of automatically detecting incidents. Please note that the SimRa data are not optimized for automated processing and [ML](#) but rather for aggregated statistics and review by humans. Therefore, several preprocessing steps are needed (Section ??). We describe our [ML](#) model in Section ?? and the training process in Section ??.

Preprocessing

To overcome some limitations of the SimRa data set, we use data cleaning and preprocessing steps, in a sequential multi-stage manner, some of which are specific to some model types that will be used afterwards to classify the incidents within rides.

Before the preprocessing phase, a typical ride can be expressed by a $n \times d$ sparse matrix $X^{(i)}$, where d describes the number of sensor features and n represents the number of timestamps in a given ride $i \in \{1, \dots, R\}$.

Note that we typically only use the accelerometer, gyroscope, and [GPS](#) sensor features. Using the linear accelerometer features in addition did not lead to a significant improvement. Furthermore, the linear accelerometer and the rotation vector features are only available in the newer Android rides. The non-sensory features such as phone

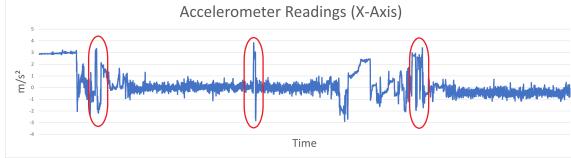


Figure 3.1: The accelerometer readings of an example ride. The encircled spots could indicate an incident but also driving over a curb.

location and bike type have a non-logical strong correlation with incidents caused by issues in the data recording phase and are therefore not used.

The preprocessing pipeline starts off with a manual label cleaning procedure that aims to remove some of the wrongly labeled incidents. Identifying them solely based on time series data is practically impossible for a human. When visualizing the accelerometer sensor readings, incidents usually stand out as sudden spikes (see Figure 3.1), but they also could be caused by driving over a curb or suddenly stopping due to a red light. Therefore we focus on the incidents that feature an additional description that was provided by the users. As it is very time-consuming, this is the only manual preprocessing step, and we apply it only to the newer Android data set. Note that this procedure did not result in a fully cleaned data set. Next, the timestamps within rides are sorted. Afterwards, we sort out invalid rides, i.e., rides that contain adjacent timestamps that have been recorded with a gap of more than 6 seconds. Furthermore, we remove outliers based on the statistical definition of outliers by Tukey et al. [87] that characterizes a data point as an outlier if it fulfills one of the equations $Outlier < q_{25} - k \cdot IQR$ or $Outlier > q_{75} + k \cdot IQR$ where the Interquartile Range (IQR) is equal to the difference between the upper and lower

quartiles [89, 100]. The k -values we are utilizing are 1.5 for GPS outliers regarding the accuracy feature and 3.0 for velocity outliers, as we have seen reasonable results for these values.

In a further step, speed is calculated from the distance between two GPS coordinates and their respective timestamp. Moreover, the accelerometer and gyroscope sensor data are interpolated to create equidistance over the whole time series. This is advisable, as unevenly spaced time series data tend to pose a problem to typical ML solutions [92]. Therefore, we up-sample to a frequency of 10 Hz via linear interpolation on uniformly generated timestamps with a 100 ms interval. That means the up-sampling factor is usually above 2. Although some argue that interpolation is a bad solution for unevenly spaced time series data in the context of TSC [28, 40], initial experiments have shown that this improves model performance. This preprocessing stage results in dense matrices $X^{(i)}$

For better convergence of the stochastic gradient descent optimizer used in the neural network, we normalize each feature individually by its maximum absolute value. This is nearly always an advantageous preprocessing step as it improves model stability [13].

Training the model on individually labeled timestamps did not appear to be a promising approach since incidents have a certain duration, which is typically longer than 100 ms, and it is highly unlikely that the user correctly specifies the label at the exact timestamp when the incident occurred. For that reason, we split our ride data into

10-second buckets, following a non-overlapping sliding window approach [68]. These buckets are then labeled in the following manner: we define a bucket as an incident bucket if any timestamp inside that bucket was labeled as an incident. Otherwise, we define it as a non-incident bucket.

Additionally, we apply a one-dimensional f -point DFT on each dimension of the accelerometer and the gyroscope sensor data contained in a bucket individually. This results in a more advanced temporal feature extraction approach that exploits the spectral power changes as time evolves by converting the time series from the time domain to the frequency domain [21].

To cope with the heavy label imbalance (e.g., $\approx 1 : 170$ on rides that have recently been recorded on Android devices) that is present in the data, we use a Generative Adversarial Network (GAN) with a CNN architecture to generate augmented data and thereby lower the imbalance gap by 10% as this has shown to produce good results in our experiments. The aforementioned f -point DFT is applied on these synthetic incident buckets as well.

Model Architecture

As our problem setting is similar to the HAR task (see Section ??), we build a customized ANN inspired by the DeepSense architecture proposed by Yao et al. [97].

In a first step, the network input is split based on the sensor that has produced it into accelerometer, gyroscope and [GPS](#) (i.e., velocity). Simultaneously, the previously Fourier transformed accelerometer and gyroscope data are separated into their real and imaginary parts.

Then, [SF](#) is applied, a method that considers each sensor individually in order to extract sensor-specific information [29]. Furthermore, it also enables the application of different individual subnets that are varying in complexity for each sensor input. Each subnet has three convolutional layers that use 64 kernels, kernel sizes between $(3, 3, 1)$ and $(3, 3, 3)$, and a stride size of 1. While in the first convolutional layer no padding is used, the second and third convolutional layers apply zero-padding which differs from the original DeepSense framework proposed by Yao et al. [97]. Another difference is that we use 3D-convolution instead of the 2D- and 1D-convolution that were applied in the original model. 3D-[CNNs](#) are more suitable for detecting spatiotemporal features compared to 2D-[CNNs](#) [85]. The described convolutional layers are complemented by batch normalization layers to reduce internal covariate shifts [45], by Rectified Linear Unit (RELU) activation, and by Dropout layers for regularization.

Our addition of residual blocks is also a slight modification of the original framework. The reasoning behind that change is that, in some cases, deeper models might have difficulties in approximating identity mappings by multiple nonlinear layers [41]. Residual blocks have been applied with great success to overcome this issue [41].

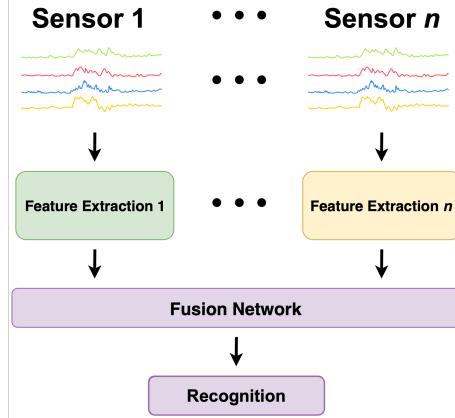


Figure 3.2: The model architecture in a nutshell: Using subnetworks for feature extraction of the different sensors (accelerometer, gyroscope, and GPS) and afterwards a fusion network to combine the features for final incident recognition [21].

Next, the outputs of the different subnets are merged in a convolutional fusion network. Its architecture is similar to the individual subnets containing six convolutional layers, residual blocks, batch normalization, RELU activation, and Dropout. The full process is shown in Figure 3.2.

The last big component of CycleSense is a RNN. RNN architectures such as LSTM [43] or Gated Recurrent Units (GRUs) [24] are capable of holding information the network has seen before and using it to make predictions in the current state. In doing so, it is possible to identify patterns or relationships inside the timestamps of a bucket or between buckets. Similar to Yao et al. [97], we also chose stacked GRU cells as they efficiently improve the model capacity [35].

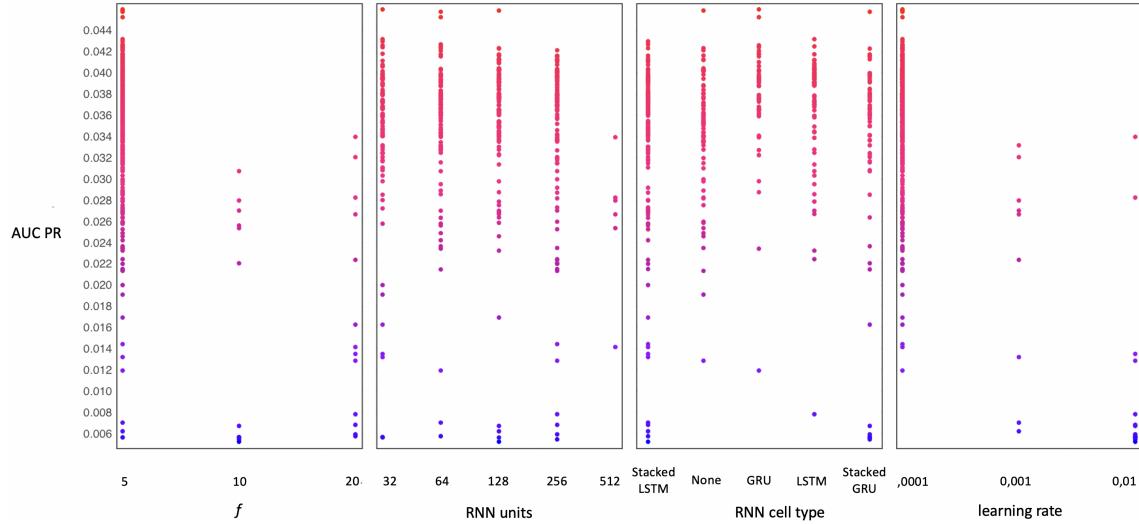


Figure 3.3: Results of the hyperparameter optimization for the four variables f (the window size), the number of RNN units used, the RNN cell type utilized, and the learning rate from left to right.

To determine the optimal set of parameters for training CycleSense, we have conducted a grid search on a variety of hyperparameters, some of which are shown in Figure 3.3.

In the following, we use GPS, accelerometer, and gyroscope data as model input if not indicated otherwise. Linear accelerometer data was only used in a few experiments as it is not available in our iOS and older Android data sets. Our implementation of CycleSense is available on GitHub⁴.

⁴<https://github.com/simra-project/CycleSense>

Model Training

For model training, the data set was split randomly into a training set (60%), a validation set (20%), and a test set (20%). Furthermore, the exact same splits are used for each model to improve comparability. We trained the final model for 60 epochs on an NVIDIA K80 GPU. We utilized a Binary Cross-Entropy ([BCE](#)) loss function that was updated with Adam optimization, a Stochastic Gradient Descent ([SGD](#)) method, a learning rate of 0.0001, and early stopping with a patience value of 10 epochs on the [AUC ROC](#) of the validation set. It is important to note that we did not use the complete SimRa data set. Instead, we used only a smaller subset of more recent rides recorded on Android devices since the heterogeneity of the data set across different versions and operating systems (see Section ??) did not allow us to train a model properly on the full data set. In addition, due to limited access to hardware, we focused predominantly on data originating from the Berlin region if not stated otherwise.

As previously described, one notable challenge we were facing was the extreme label imbalance present in the data. This is due to the fact that incident buckets are far rarer than non-incident buckets. To cope with that, we trained our model by using a weighted loss function with the class weights of the train data set as weights. For example, the class weights were 1 and 170 for the rides that have recently been recorded on Android devices.

While the DeepSense model was trained in a standard fashion, we use stacking during the training of CycleSense. Stacking (or stacked generalization) is an ensemble learning method that combines the predictions of several different models in order to contribute equally to a collective prediction.

However, we are also not interested in equal contributions of the network since that could overvalue models with a poor performance. We therefore changed the CycleSense model to an integrated stacking model by adopting the idea of stacked generalization [93], where the fusion network acts as the meta-learner. Also, we deviate from a pure stacking model. This is the case, as the meta-learner does not get any classification output of the subnetworks as input aside from the latent features in the last layer of the subnetworks. Thereby, the weights of the submodel layers that have been pretrained individually are loaded and frozen, so they are not updated during the training of the whole CycleSense model. This learning procedure further improved our results as shown in Section ??.

3.1.2 Evaluation of CycleSense

To evaluate CycleSense' training results, we have to put them into context. For this purpose, we compare them to the two detection methods currently used in the app as discussed in Section ?? . We give an overview of the changes we made to the baseline methods with the goal of a fair comparison in Section ?? . We also describe

the metrics that we use to compare our model to the baseline methods (Section ??) before presenting the results of our evaluation (Section ??).

Baselines

The first baseline is our original heuristic [50] which is based on the underlying assumption that incidents will often result in sudden acceleration spikes, e.g., when braking or swerving to avoid obstacles. We made some small changes to this heuristic to enable its compatibility with the [AUC ROC](#) metric, thus, increasing the comparability with our approach.

As a second baseline, we retrained the Fully Connected Network ([FCN](#)) model from the alternative approach [75]. We used the original preprocessing pipeline (which differs significantly from the here presented one) but used the full data set as introduced in Section ???. We skipped the under-sampling step, disregarded the phone location and the bike type feature for the reasons mentioned in Section ??, and used a non-overlapping sliding window approach with 10 second windows for better comparability.

The third baseline is DeepSense [97], which we implemented and trained as the authors describe in their work. For the differences between DeepSense and CycleSense, see sections ?? and ??.

Based on these changes for improved comparability, we retrained the original model. We use both baselines for comparison as they are, to our knowledge, the only approaches for (semi-)automatically detecting incidents based on sensory time series data. Furthermore, they have been developed on the SimRa data set, which enables a fair comparison.

Metrics

Due to the massive label imbalance already mentioned earlier, common metrics such as accuracy, F1-score, and precision are difficult to interpret. Moreover, in our scenario it is more important to find the true near miss incidents than to classify non-incidents correctly, as False Positives can be more easily corrected by the user of the SimRa app. For both reasons, a high number of False Positives is more acceptable than a low number of True Positives, which further limits the usefulness of such metrics like precision, F1-score, or Matthews correlation coefficient ([MCC](#)). Therefore, we focus on the [AUC](#) of the [ROC](#) metric, which is insensitive to changes in class distribution [\[31\]](#) while also reporting the respective confusion matrices.

Evaluation Results

In a first step, we compare CycleSense to the two baselines and common model architectures used in the context of [DL](#) for [TSC](#) [\[46\]](#). All of these were trained on

the Android data set consisting of more recent rides which provides the best results for all approaches. Figure 3.4 and Table 3.1 show the differences in performance.

The FCN and CycleSense clearly outperform the modified heuristic (0.621 AUC ROC). However, there is still a big performance gap between our model and the FCN model. While the FCN model scores 0.847 AUC ROC, CycleSense achieves an AUC ROC score of 0.906, i.e., there is a chance of $\approx 90.6\%$ that the model can distinguish correctly between a randomly chosen incident and non-incident bucket. Furthermore, our model performs better than other model architectures that are common for DL in TSC [46]: Auto Encoder, Gramian Angular Field (GAF), Echo State Network (ESN), and the CNN-LSTM model. With regard to the increasing model complexity, we clearly see diminishing returns. We can see this in the example of the rather simple CNN-LSTM model which exhibits a relatively close performance to the much more complex CycleSense model with stacking. The CNN-LSTM model has $\approx 90,000$ parameters, while the CycleSense model has $\approx 1,100,000$ parameters. As a consequence, the time to evaluate the test set of the newer Android data consisting of 795 rides took 4 seconds with the CNN-LSTM model and 56 seconds using CycleSense on the NVIDIA GPU.

In another experiment, we include the linear accelerometer sensor values in addition to the accelerometer, gyroscope and GPS data we used so far. The result for CycleSense is again an AUC ROC of 0.906, although the model requires more memory, training and processing time. Therefore, we leave out the linear accelerometer feature.

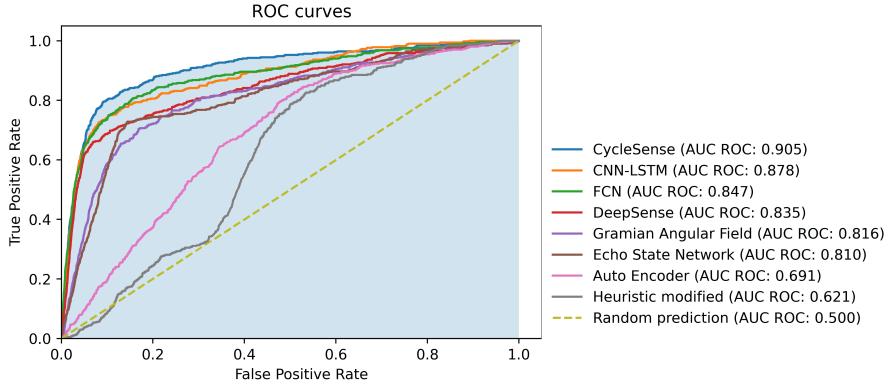


Figure 3.4: Comparison of the baselines and common model architectures used in the context of [DL](#) for [TSC](#) [46] with the CycleSense model. Note that all of these models have been trained and evaluated on rides contained in the SimRa data set that have been recorded with more recent versions of the SimRa Android app.

So far, we have predominantly focused on newer rides recorded on the Android version of the SimRa app. As shown in Figure 3.5a, the model performs far worse on the other splits of the data set. Therefore, it was necessary to train an individual model for each part of the data set. This yields far better results (Figure 3.5b).

As the Berlin region has by far recorded the most rides, we have so far used only those for tuning, training, and evaluating our model. The SimRa app, however, is deployed in many more regions, so our model is clearly required to perform there, too. For this reason, we have evaluated the Berlin CycleSense model on the newer Android rides coming from Hanover and Nuremberg.

The outcome of this experiment is visualized in Figure 3.6a. It clearly shows, that CycleSense does not perform as good as in Berlin. Since most regions lack training

	TN	FP	FN	TP	AUC ROC	Precision	Recall	F1-Score	MCC
CycleSense	107934	10893	104	400	0.906	0.035	0.794	0.068	0.156
CNN-LSTM	106427	12400	127	377	0.878	0.030	0.748	0.057	0.135
FCN	100932	18500	98	402	0.847	0.021	0.804	0.041	0.115
DeepSense	106192	12635	153	351	0.835	0.027	0.696	0.052	0.123
GAF	98782	20045	150	354	0.816	0.017	0.702	0.034	0.092
ESN	101670	17157	138	366	0.810	0.021	0.726	0.041	0.107
Auto Encoder	58416	60411	88	416	0.691	0.007	0.825	0.014	0.041

Table 3.1: Comparison of the baselines and common model architectures used in the context of [DL](#) for [TSC](#) [46] with the CycleSense model. See the discussion in Section ?? about the usefulness of various metrics. Note also that all of these models have been trained and evaluated on rides contained in the SimRa data set that have been recorded with more recent versions of the SimRa Android app. For all models the threshold which optimizes Youden’s index[98] were chosen.

data, it would not be a feasible solution to train models individually per region in the current stage of the SimRa project. Instead, we retrain CycleSense on a data set that includes all the rides recorded on the newer versions of the Android app within the Berlin, Nuremberg, and Hanover regions. The results from Figure 3.6b show that this clearly improves the performance in these additional regions. At the same time, the performance on the Berlin data set has declined only slightly (0.029 [AUC ROC](#)) by comparison. Nevertheless, the [AUC ROC](#) for Berlin is still the highest and clearly above Nuremberg, which is well ahead of Hanover.

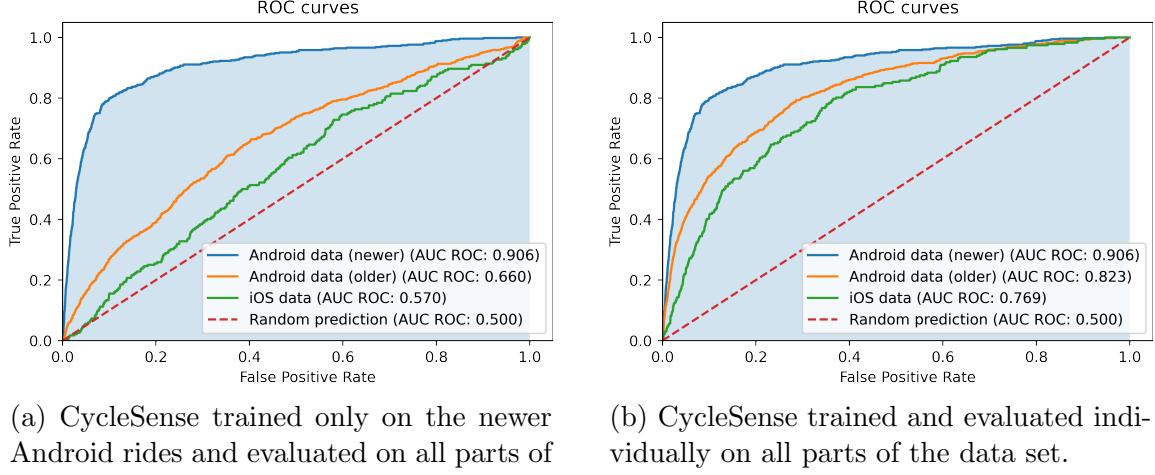


Figure 3.5: Comparison of CycleSense trained only on the newer Android rides or individual CycleSense models trained for each part of the data set. Note that only the newer Android data set was manually cleaned.

3.1.3 Discussion of CycleSense

Our results demonstrate that the proposed CycleSense model outperforms every other model that we have compared to. While improving the model’s architecture and other components, we identified a number of challenges inherent to the problem setting that we discuss in the following.

Impact of Preprocessing & Training Steps

In a first step, we want to highlight the importance of our preprocessing and training methods for the success of our model. Therefore, Figure 3.7 illustrates the perfor-

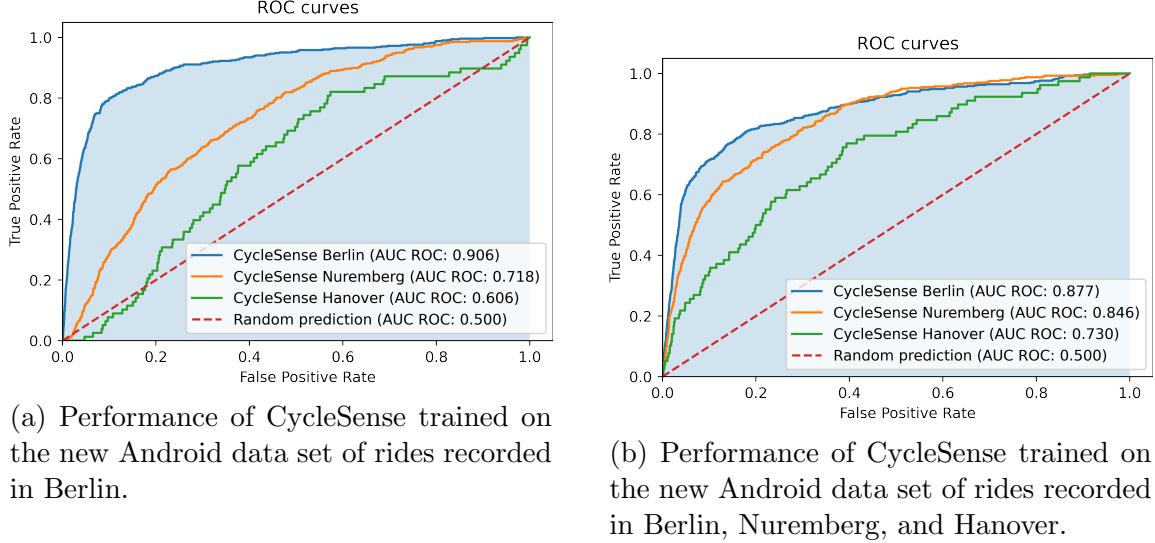


Figure 3.6: Comparison of the performance of CycleSense trained on Berlin and trained on the other regions combined.

mance of CycleSense when one of the preprocessing or training methods is skipped, resulting in a significant drop in performance in each of the four examples.

Limitations of Crowdsourced Data

Crowdsourced data are generally noisy which could contribute to a reduced model performance. They do also contain several biases, e.g., the Selection Bias, Device Positioning Bias, Extreme Aversion Bias, or Confirmation Bias [7, 19, 48]. In this context, the heterogeneity across devices and platforms is likely another factor of influence. As already described, the application that is used by contributors to record their rides is available on two platforms. Those platforms are supported by

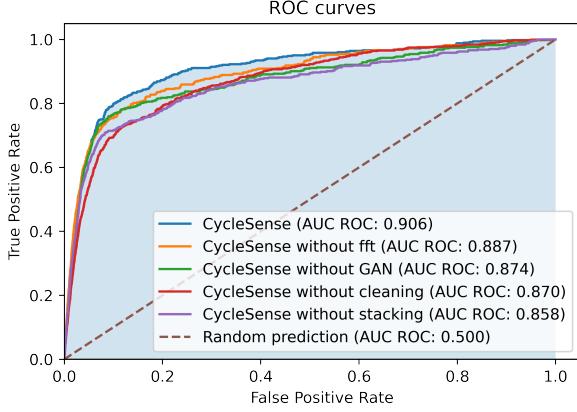


Figure 3.7: Impact of different preprocessing and training steps on the performance of CycleSense.

a wide range of different devices and models with different hardware inside. Phone manufacturers use different [GPS](#), gyroscope, and accelerometer sensors that can vary immensely in sensitivity and overall behavior. Stisen et al. [79] have shown that there is major heterogeneity when it comes to the accuracy of sensor readings. While the main focus of that study was on accelerometer data, Kuhlmann et al. [55] have shown that orientation sensor data is also affected by this variability. This is in contrast to the [HAR](#) tasks, we compare our task to. The [HAR](#) data set [4] was generated under laboratory conditions that always used the same smartphone type body mounted to the exact same position on selected study participants. This significantly simplifies the classification task. Another factor that contributes to the issue of noisy crowdsourcing data in the SimRa data set is the fact that some users misinterpret the purpose of the SimRa app. Instead of labeling incidents, they report, e.g., dangerous areas or annoying traffic lights. These can of course not be

captured by the sensors used. Many such “incidents” can be identified through the comment column of the data set. It should also be noted that the SimRa data set was not recorded and labeled with the goal of developing ML models – the goal was to record data which will be analyzed and processed by humans.

Technical Limitations of the SimRa Data Set

Recording rates of sensor readings deviate among devices and operating systems and impair model predictions [79], this may also affect the performance of CycleSense: As illustrated in Figure 3.8, the recording rates differ significantly in older and newer Android rides as well as in iOS rides. While the iOS rides’ median (≈ 300 ms) is similar to the one of newer Android rides, the IQR is much greater and spans approximately 150 ms. This circumstance could indicate that the relatively weak model performance on this data set can partly be explained by that factor. Furthermore, the gyroscope data is recorded with a higher frequency in newer Android rides than in older Android or iOS rides. This factor could also contribute to the different model evaluation results. The achieved AUC ROC for uncleaned newer Android data was 0.870, while it was 0.823 for the older Android data.

Aside from that, the moving average that is used in the SimRa app to condense the data and comply to users’ upload volume constraints [50] reduces the amplitude and shifts the exact point in time of incidents and other events. This has the effect that incidents and non-incidents are hard to distinguish as illustrated in Figure 3.9. Both

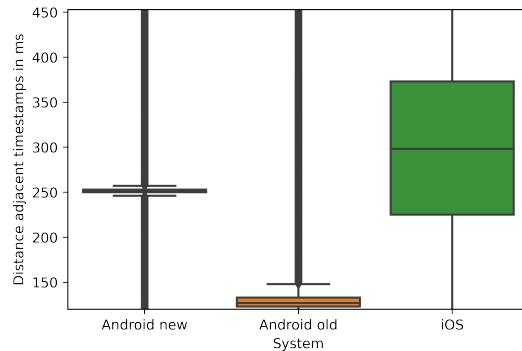


Figure 3.8: Box plot showing the empirical measurement times observed in the SimRa data set for rides recorded with different versions of the SimRa app.

these factors could hurt the model’s ability to classify incidents correctly based on the data. It is important to acknowledge that this issue becomes less severe when the sampling frequency is higher, as it is the case for the older Android data.

Limitation of the Classification Task

There is an inherent label imbalance present in our data set. Only relatively few rides contain incidents. Moreover, the ratio between timestamps that represent an incident and those that do not is even smaller. As a consequence, there is an enormous imbalance between the different label categories.

Furthermore, some incident types such as tailgating or close passes might not be detectable at all in accelerometer and gyroscope sensors [3, 50] as cyclists might not change their motion profile despite (or even due to) a dangerous situation. For detecting these kinds of incidents it may be necessary to include different sen-

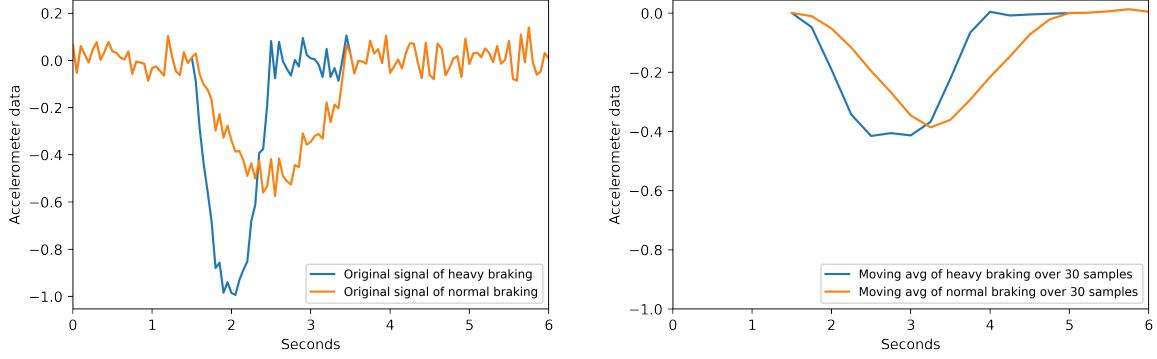


Figure 3.9: Visualization of the sensor data of a simulated heavy braking incident vs. a moderate braking event before and after the moving average has been applied.

sors. In the SimRa project, this is already happening through an integration of OpenBikeSensor ([OBS](#))⁵ data which measures the passing distance of cars. The current data set, however, only includes very few OBS-supported rides.

In contrast to our classification task, the inherent classification problem of the [HAR](#) data set [4], is to classify reoccurring ongoing patterns such as walking or jogging, while an incident might need to be detected by one short non-reoccurring event such as sudden braking. This further complicates the matter.

Data Set Shift

A major challenge in the context of crowdsourced data are data set shifts. Depending on the device (see also ??), but also on other factors, such as the city of origin of

⁵<https://www.openbikesensor.org/>

the recorded data (see also ??), there could be non-stationarities in the data that violate the assumptions underlying almost all ML models and thus could impact the generalization performance of the trained models. Several types of data set shifts can be distinguished, including *covariate shifts*, meaning shifts in the input data [81], *label shifts*, meaning shifts in the distribution of targets [59], or mixtures thereof. Detecting these shifts [1, 8, 16, 71, 72] and predicting [77] or reducing their impact on the generalization performance is an active field of research [12, 78]. Some approaches aim at model specific improvements to alleviate data set shift [81]. These approaches have a decisive disadvantage, most of these approaches only work for one model class and require access to the inner workings of the ML pipeline, often after the feature extraction step. More promising and easier to build and maintain are model agnostic solutions that focus on the data, rather than the models, to detect and counteract data set shifts [12]. Extending the training data sets to account for all variation and shifts in the data that the ML model should be invariant to, often called *augmentation*, is a popular and effective way of counteracting data set shift, see for instance [25]. In our work we follow this line of thought of a data centric AI approach.

3.1.4 Related Work of CycleSense

One of the most common tasks for TSC are Natural Language Processing, speech recognition, and audio recognition in general. Another field that deals with this prob-

lem is Human Activity Recognition ([HAR](#)) which is concerned with identifying the specific activity of a human based on sensory time series data. Time windows of a few seconds are classified into activity categories (e.g., walking, sitting, running, lying). While various feature extraction and pattern recognition methods have been successfully applied in the past in this context [17], those approaches have constraints such as hand-crafted feature extraction, being able to only learn shallow features [96], or the requirement for large amounts of well-labeled data for model training [91]. Deep Learning techniques and more specifically [CNNs](#) have recently proven to overcome these issues and deliver convincing results in the context of [HAR](#) [73, 91]. Also, implementations of [RNNs](#) such as [LSTM](#) [83, 97] have proven to be successful. Yao et al. [97] use a combination of [RNNs](#) and [CNNs](#) on top of a [SF](#) approach after preprocessing their input data using a [DFT](#).

3.1.5 Summary of CycleSense

An increased modal share of bicycles is necessary for solving emission and traffic related urban problems. A key challenge for this, is the lack of (perceived) safety for cyclists. Improving the situation requires detailed insights into safety levels of street segments – the SimRa platform [50] has been proposed as a data gathering mechanism for incidents and cycling tracks. While this is an important step towards data collection, the platform relies on manual annotation of tracks which limits the number of potential users.

In this paper, we have proposed CycleSense – a model for automatic detection of such incidents. Using the SimRa data set, we have shown that CycleSense is capable of detecting incidents on the basis of accelerometer and gyroscope time series data in a real-world scenario. It can correctly distinguish between an incident and a non-incident with a probability of up to 90.5%. We have also compared it to the heuristic currently used in the SimRa platform and an existing [FCN](#) that was specifically developed for SimRa. Additionally, we have implemented several [DL](#) models that are frequently used in the context of [TSC](#). We were able to show that our model outperforms all of these approaches.

While this is an important step towards fully automatic incident detection, we believe that – in the context of the SimRa platform – CycleSense should be complemented with human annotation in a semi-automated way for the foreseeable future. Although this does not quite reach our long-term goal of full automation, it should significantly decrease the annotation effort and should also lead to improved labeling quality. In the future, this could, in turn, be used to further improve CycleSense. Since April 2022, the SimRa app has been using CycleSense for incident detection.

3.2 CycleQuality

Cities all over the world aim to increase the modal share of bicycle traffic, e.g., to address emission problems, frequent traffic jams, but also to improve the citizens'

health through more daily activity. Key factors for this are traffic safety, topography but also the comfort level of cyclists [32, 50, 66]. All these factors are affected by the availability and quality of cycling infrastructure which needs to be monitored and maintained continuously. Even monitoring of surface quality alone, however, can be challenging due to the sheer dimensions of existing infrastructure. Germany’s capital Berlin, for example, has around 2,300 km of bicycle tracks as of March 2023, i.e., manually inspecting cycling infrastructure to monitor surface quality is infeasible.

To address this problem, we propose to use a crowdsensing approach in which cyclists record their daily rides using a smartphone app. Using the built-in motion sensors, the surface quality (or rather the lack of) can be measured as vibrations or bumps. In a second step, we can then combine data from multiple rides to derive an estimate for the surface quality. This way, monitoring of surface quality can be automated to a high degree at little cost and the resulting data can be used for maintenance planning or surface quality-aware routing. Especially for highly frequented cycling tracks, surface quality problems can be detected quickly.

While there are other projects studying the surface quality of cycling infrastructure through crowdsourcing (e.g., Luedemann et al. [61]), our work is unique and novel because alternative approaches (i) either focus on categorizing the surface *type* (e.g., cobblestones) where we focus on the surface *quality* (i.e., the level of roughness) or (ii) have strict assumptions on phone positioning and other properties where we rely on a “wisdom of the crowds” strategy to filter out noise. The latter is also an advantage because, unlike related approaches, which use a dedicated app for measuring the road

surface quality, thus, potentially forcing cyclists to run multiple apps, our approach can easily be retrofitted to existing apps and can even be used to analyze already stored datasets.

For this, we make the following contributions:

- We present a data processing pipeline for deriving surface quality which combines signal processing with geographical clustering techniques in the form of edge-based preprocessing on the phone and cloud analytics (??).
- We describe how we integrate this data processing pipeline in the existing crowdsensing project SimRa [50] which up to now has only focused on traffic safety(??).
- We describe how the resulting data can be used to increase the comfort of cyclists through surface quality-aware routing and how the data can be exposed to city administrations(??).
- We evaluate our approach by analyzing the SimRa dataset [10, 11, 49] and comparing it on-site conditions in eight streets with different surface quality (??).
- We discuss to which degree our approach can automate surface quality monitoring and how additional sensors could possibly help to improve data quality (??).

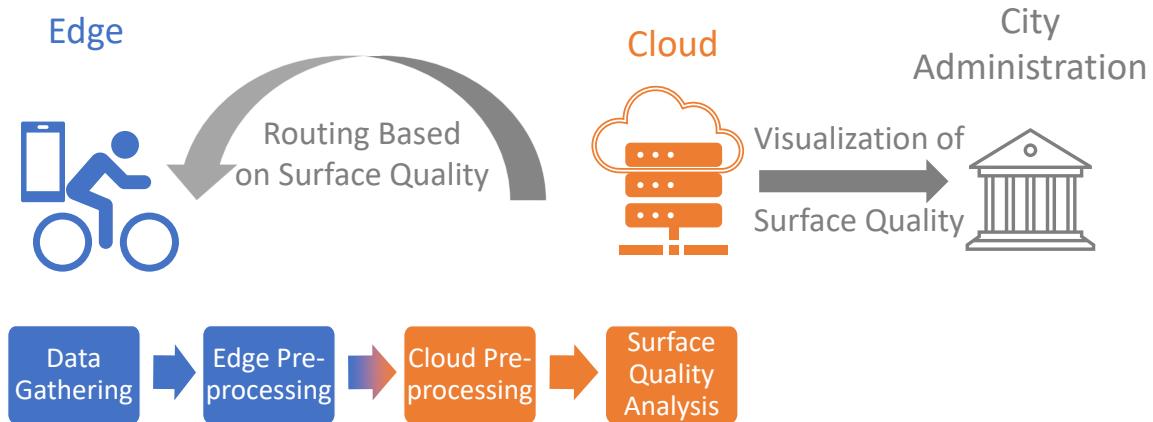


Figure 3.10: Overview of our surface quality analysis pipeline.

3.2.1 Data Processing Pipeline

In this section, we start by giving a high-level overview of our data processing pipeline for deriving surface quality ?? before describing the individual steps in the process (?? to ??).

Overview of Pipeline

The tasks of our data processing pipelines are distributed over the edge (i.e., on the smartphone) and the cloud, see also Figure 3.10. After collecting data in the SimRa app using the built-in motion sensors, data is preprocessed locally before upload to our backend servers. There, additional preprocessing steps are executed before the actual surface quality analysis.

The parts of the preprocessing which reduce the amount of data need to be run on the edge to preserve user privacy and to reduce bandwidth consumption. The data cleaning parts are too compute-intensive and are executed in the cloud to reduce power consumption on the phone.

Preprocessing in the Cloud

The preprocessing in the cloud has two main goals: cleaning noisy data and calculating a value that represents the smoothness of the ride. For the first goal, we cut off ten seconds each at the start and end of the ride to remove noise resulting from users (un)mounting their bikes or putting the phone in a pocket. Ideally, this would of course already be done on the phone but could not be done here since the main use case of SimRa requires this data, i.e., we would have had to upload the data twice. We also remove stops, i.e., only the parts of the ride which have a minimum speed of 5 km/h and a minimum duration of 1 minute are considered. Such low speeds usually occur, when the bicycle is being pushed by the cyclist rather than being ridden or when stopping at traffic lights. Even if the cyclist should manage to ride so slowly, the ride will be prone to unsteady motions, leading to noisy data. This also removes parts where users forgot to stop the recording after their ride.

For the second goal, we normalize the accelerometer data by calculating the mean of the three moving variances (window size 10) of the axes X, Y and Z. This preserves information about amplitude but disregards the direction which is highly affected by

the position of the phone. Thus, we obtain a time series of single values representing the bumpiness of the surface in each in-motion ride segment.

Surface Quality Analysis

Each cyclist creates a different data track on a given road, and without calibration, it is not possible to accurately analyze each track individually. Ideally, we would at least use cyclist-specific profiles (i.e., per cyclist aggregates) which, however, are not available due to privacy reasons (rides are pseudonymized individually [50]). The idea behind our approach is to take advantage of the size of the data set and use the law of large numbers to obtain robust results without having to calibrate the data and without affecting the results too much by noise.

For this, we consider in the first step each ride individually: We take the preprocessed ride (which is a single time series of aggregated motion sensor readings without stops plus the GPS trace) and calculate the percentiles (0.2, 0.4, 0.6, 0.8, and 1) of the time series. For normalization, we then replace all motion sensor readings with values 1 to 5 depending on which interval they fall into, i.e., a motion sensor reading from the interval (0.2; 0.4] would be replaced with the value 2. The intuition behind this is that longer rides are likely to encounter very different surface quality, hence we calculate the relative bumpiness of an area in comparison to the rest of the ride's bumpiness.

In the second step, we use the data from all such normalized rides and, using the GPS trace, map them to a grid of $10m^2$ cells. For each cell, we hence have a distribution of values 1 to 5. As a metric for the bumpiness of that cell, we use the average of all values – e.g., when color-coding a map – but make other statistical metrics available as well (see, e.g., the distribution function charts in our evaluation section).

3.2.2 Using Road Surface Quality Information

In this section, we describe how such surface quality results can be used. In ??, we describe how we implement a navigation feature into the SimRa app that uses the road surface quality as an additional parameter in routing. We then show in ?? how we can expose the output data of our pipeline to city administrators.

Routing with Surface Quality

With the surface quality scores calculated, it is possible to provide a route planning feature, where not only distance and time are considered, but also the surface quality. The main question here is how much the surface quality should be weighed when calculating the best route from A to B or rather what detour lengths are acceptable. We decided to give the user the opportunity to influence this factor with the usage of a slider, that can be set between 0 for not considering surface quality in the routing and 10 for the highest importance of the surface quality (fig. 3.11). We host

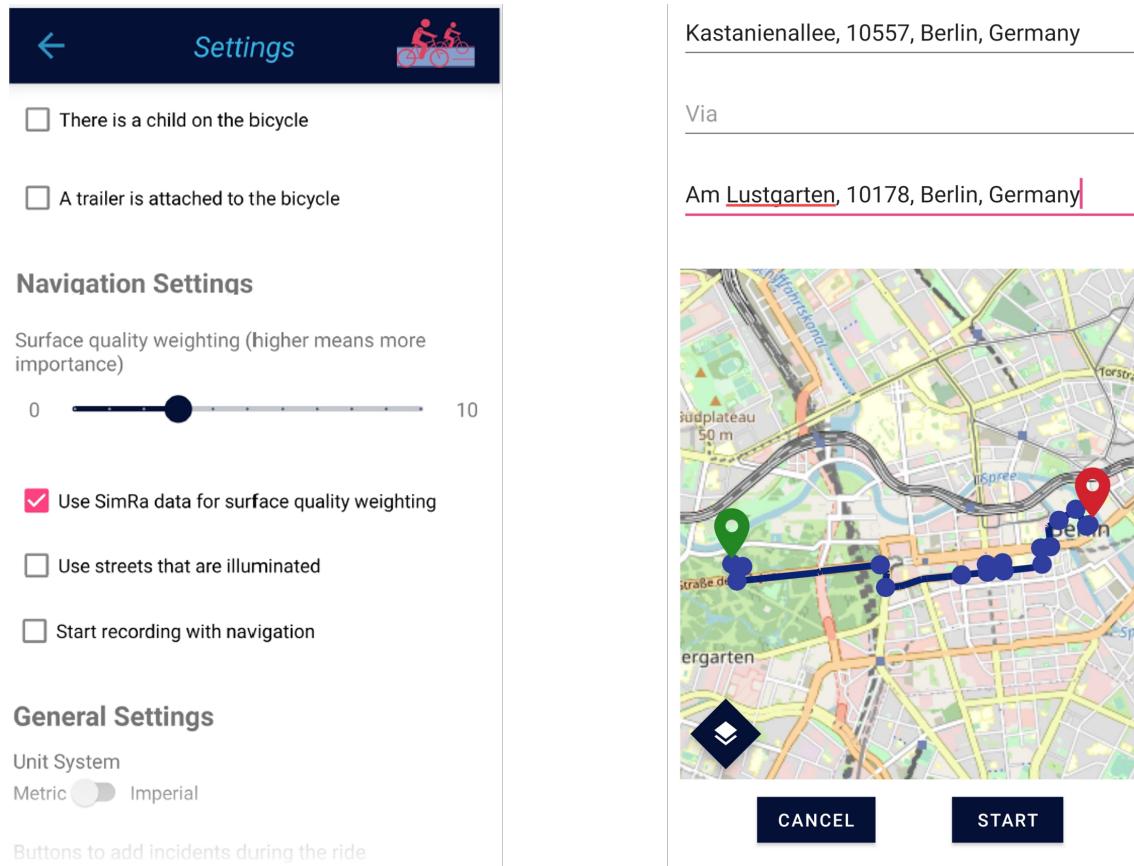


Figure 3.11: With a slider in the settings menu, the weight of the surface quality in the routing can be set.

a modified GraphHopper⁶ for routing. GraphHopper uses edges for streets, that are connected via nodes. Each has a weight for routing purposes and it is possible to change the weight according to custom data. This is where we use the surface quality by increasing the weight depending on surface quality and user-specified influence factor.

Output Data and Visualization

Our bicycle road surface quality pipeline creates a GeoJSON file as an output. It contains the cells with a surface area of $10m^2$ in a grid as **Features** of the **geometry** type **Polygon** with the surface quality score information such as mean, median, standard deviation, number of rides going through the cell and coloring information in the **properties** key. With such an output file, it is very easy to create a simple visualization⁷, e.g., with Leaflet⁸, as depicted in fig. 3.12. Using a visualization like this, also non-tech-savvy users can easily monitor the bicycle road surface quality of a large area and take action where needed.

⁶<https://github.com/graphhopper/>

⁷<https://simra-project.github.io/surfaceQuality/Berlin.html>

⁸<https://leafletjs.com/>



Figure 3.12: A visualization of the output file showing the surface quality of the boxes when hovering over them with the mouse cursor. Color coding is based on the average value.

3.2.3 Evaluation of CycleQuality

In this section, we evaluate the surface quality analysis by comparing the calculated surface quality of selected street segments across Berlin with their surface type in the real world, which we get from OpenStreetMap (OSM). As data input, we use the existing SimRa datasets [10, 11, 49] (almost 90,000 rides with more than 650,000km in total). Based on the intuition that different surface types will also be partially correlated with surface quality, we randomly picked four spots with different surface types. To also show the limitations of our approach, we then explored the dataset and manually picked four additional spots where our approach appears to have returned the wrong results. Each evaluated section has a surface area of 10 m² and to compare them to each other we analyze their mean, median, and standard deviation values.

We first describe the clear results from the first group (??). Afterwards, we categorize the additional four “problem spots” as mixed results (??) or seemingly incorrect results (??).

Sections with Clear Results

We evaluate at least one example for each of the following surface types, which are sorted in descending order with regard to their expected surface quality [84]: asphalt, flat paving stones, fine gravel, cobblestones. We chose the sections in a way that (i)

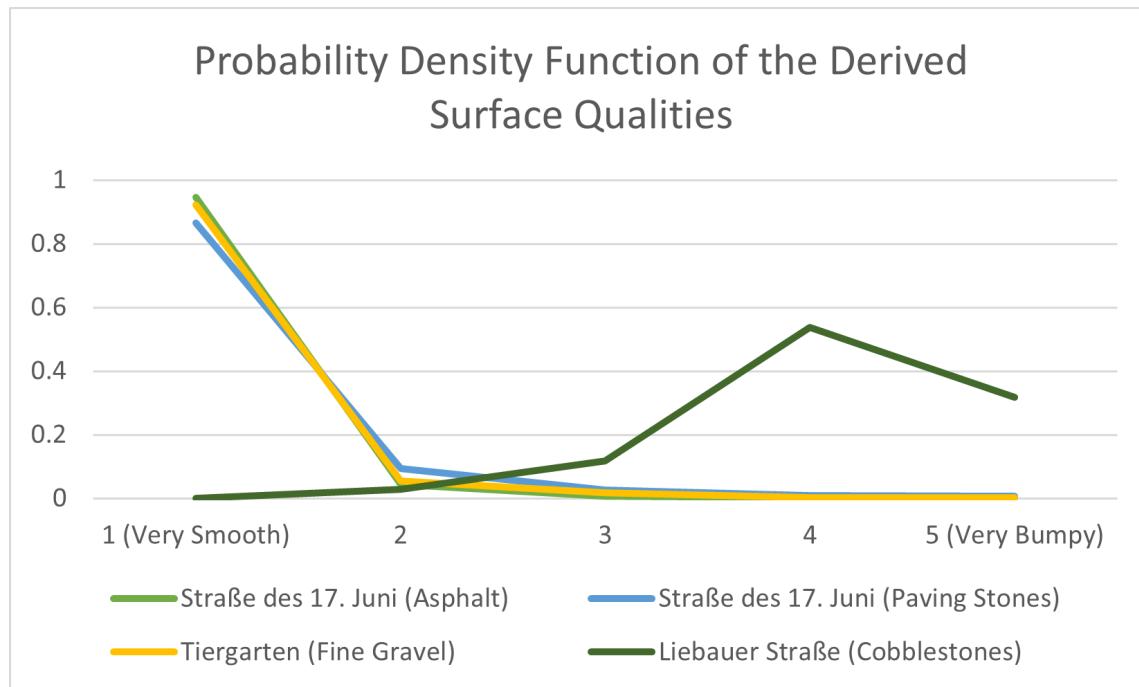


Figure 3.13: The Probability Density Function of the Derived Surface Qualities shows that these segments have very undisputed road surface quality values, since they are either very good or very bad.

asserted that we have sufficient data for them and (ii) to cover all different surface types. After filtering based on these criteria, we randomly picked four sections.

table 3.2 and fig. 3.13 show the results of the surface quality analysis of the selected sections with very clear and intuitive results.

It can be observed, that the aforementioned list of surface types, which was sorted in descending surface quality order, was ordered correctly. A newly maintained asphalt section in *Straße des 17. Juni* has a nearly perfect score, which means that its surface

Table 3.2: Surface Quality Analysis Evaluation Results Showing Mean, Median and Standard Deviation of Sections With Clear Results

Street Name	Surface	GPS Location	Mean	Median	Std. Dev.
Straße des 17. Juni	Asphalt	52.515369,13.630855	1.03	1	0.17
Straße des 17. Juni	Paving Stones	52.513501,13.335127	1.25	1	0.5
Tiergarten	Fine Gravel	52.514745,13.34622	1.32	1	1.06
Liebauer Straße	Cobblestones	52.509132,13.453806	4.15	4	1.07

quality was in the top 20% in almost all rides crossing this section. Followed by that are two sections with flat paving stones and fine gravel as their surface type, which have very similar results. This means, that both flat paving stones and fine gravel have comparable surface quality in terms of bumpiness from the perspective of a cyclist. However, it should be noted, that fine gravel can be less favorable in areas with a lot of precipitation (more on that in ??). Not very surprisingly, the *Liebauer Straße* has very bad surface quality scores, since it is paved with cobblestones and has very busy sidewalks with restaurants and cafes, which prevent cyclists from (illegally) cycling there instead of on the street.

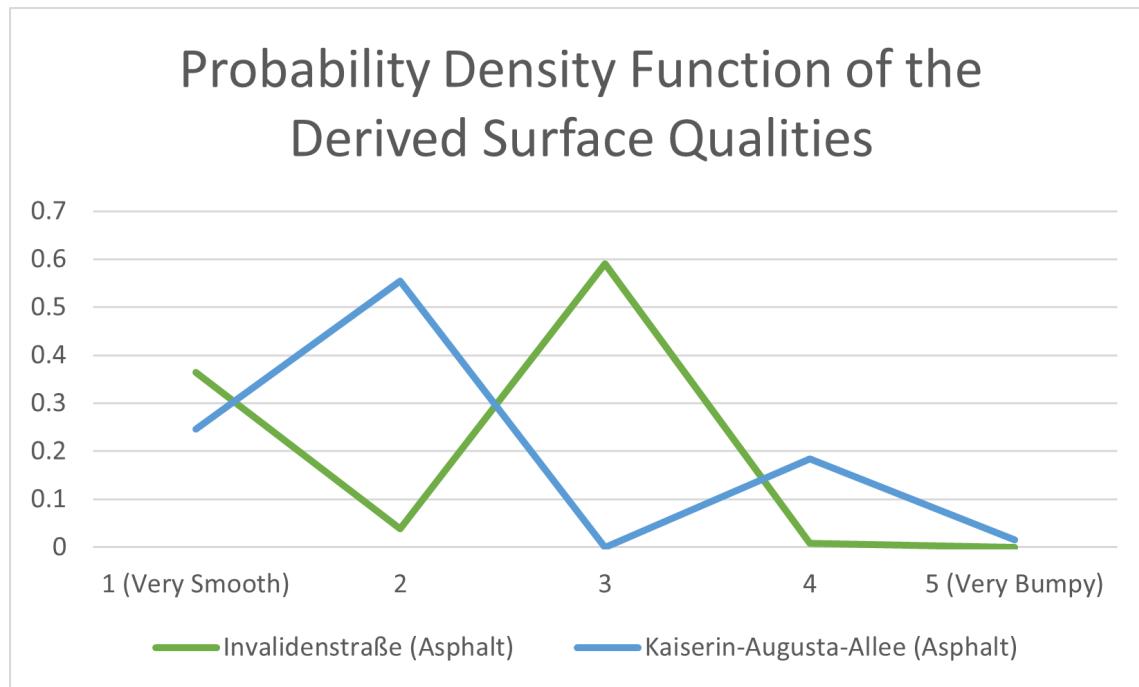


Figure 3.14: The Probability Density Function of the Derived Surface Qualities shows that these segments have confusing road surface quality values: Depending on the ride, they appear to have either very good or very bad surface quality (multiple peaks).

Sections with Mixed Results

While most results are intuitive, e.g., new bicycle roads paved with asphalt and separated from the motorized vehicle traffic having very good surface conditions, some other sections (see fig. 3.14) seem confusing at the first glance.

Table 3.3: Surface Quality Analysis Evaluation Results Showing Mean, Median and Standard Deviation of Sections With Mixed Results

Street Name	Surface	GPS Location	Mean	Median	Std. Dev.
Invalidenstraße	Asphalt	52.526236,13.369196	2.24	3	0.96
Kaiserin-Augusta-Allee	Asphalt	52.524429,13.327253	2.17	2	1.05

According to the Probability Density Functions (PDFs) of street sections in the *Invalidenstraße* and *Kaiserin-Augusta-Allee*, there seem to be two distinct surface qualities in each section. A closer look into the specific sections reveal the causes:

In *Invalidenstraße*, there are a bus lane, a bus stop, and an on-curb bike lane (see fig. 3.15). Most bus lanes in Berlin can be legally used by cyclists, however, some cyclist may still prefer the relatively bumpy bike lane. Additionally, when a bus stops at the bus station, cyclists that used the bus lane might overtake the bus on the very smooth car lane to the left. These two factors most certainly are the reason for the distinct two-peak distribution of recordings in that spot.

A closer look at the near-miss incident data in *Kaiserin-Augusta-Allee* indicates that people use the bike lane in one direction and the street in the other direction [10, 11, 49] as one side seems to be in an unusable condition while the other is acceptable. Since the bicycle lane is paved with paving stones and the street with asphalt, this leads to different surface quality of the road, depending on which direction the ride



Figure 3.15: A bus lane, a bus station and a bicycle lane on the sidewalk can create different results in a small area. (Source: Apple Maps)

was. Both directions, however, regularly end up in the same 10x10m box as the street is not overly wide, especially considering GPS accuracy.

Sections with (Seemingly) Confusing Results

There are also sections where the surface quality result seem to be completely wrong, when compared to the OSM surface type labels. Table 3.4 shows two sections, that are – according to OSM – paved with cobblestones and without separate bike lanes, but with very good results, which is confirmed by the PDFs depicted in fig. 3.16.

The possibility, that this section has a wrong label and is in fact, not paved with cobblestones may come to mind. However, as fig. 3.17 reveals, both streets are indeed paved with cobblestones.

Figure 3.17 also shows very likely reasons why these sections have very smooth bicycle rides. The infrastructural conditions incentivize cyclists to prefer the sidewalks over the actual streets. First, the cobblestones on the streets make it very unpleasant for cyclists to cycle on them. Second, the sidewalks are wide, quiet, and paved with large flat-surfaced paving stones. Third, the streets are contested by parking cars, or by cars searching for a parking spot, service vehicles, construction vehicles, etc. which will often block cyclists and also create safety hazards. Considering these circumstances, it is not surprising to see good surface quality values in these two and other similar streets as cyclists are likely to (illegally) use the sidewalks instead. In

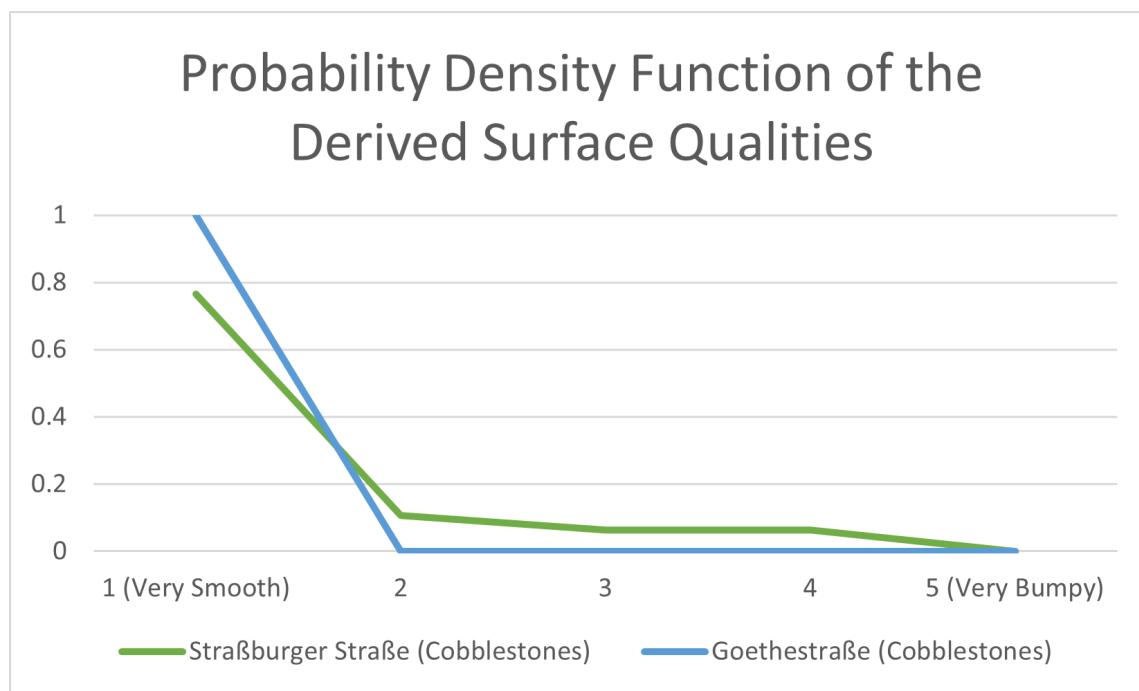


Figure 3.16: The Probability Density Function of the Derived Surface Qualities shows that these segments have very smooth road surfaces, although they are paved with cobblestones.

Table 3.4: Surface Quality Analysis Evaluation Results Showing Mean, Median and Standard Deviation of Sections With (Seemingly) Confusing Results

Street Name	Surface	GPS Location	Mean	Median	Std. Dev.
Straßburger Straße	Cobblestone	52.532273,13.416521	1.43	1	0.87
Goethestraße	Cobblestone	52.508889,13.308333	1	1	0

fact, photos on Apple Maps (not included in this paper) actually show cyclists using the sidewalk instead.

3.2.4 Discussion of CycleQuality

Overall, the results presented in this paper show, that our approach can derive surface quality information using data recorded in the SimRa app. Nevertheless, it still has a number of limitations which we discuss in this section.

Methodological Challenges

Since the SimRa dataset consists of crowdsourced data generated by smartphones, the recorded data is very heterogeneous. This comes from the fact that different smartphone models have different GPS modules and motion sensors, which leads to the problem that a road can be very smooth according to one smartphone and very

rough according to another. This problem is further aggravated by the wide range of different bicycle types, e.g., racing bicycles or mountain bikes, which in turn also heavily influence the vibrations the smartphone can sense. To solve these problems, we rely on the *law of large numbers* and compare how a section’s surface quality was relative to the ride.

Preprocessing in SimRa

As input for our surface quality analysis pipeline, we used the SimRa dataset, since it is to our knowledge the only public dataset containing a very high number of anonymized individual rides in a non-aggregated form. However, it is important to note, that SimRa was developed for identifying near miss incident hotspots in bicycle traffic. For that, a relatively low accelerometer sampling rate of 50 Hz is used before further reducing the level of detail by calculating the moving average with a window size of 30 and then taking every fifth value. While with this, it is possible to automatically detect near-miss incidents [52], a higher resolution of data points would allow us to develop a more sophisticated measurement approach. This would, however, further increase the disk space, memory, battery and bandwidth usage of the SimRa app and as we have shown in ??, the surface quality we derive are sufficiently precise for our intended use cases.

Behavior of Cyclists

Due to the GPS inaccuracy, it is impossible to identify whether a cyclist used the actual road or the sidewalk. For instance, as discussed in the previous section, fig. 3.17 showing *Straßburger Straße* in Berlin, would be expected to have poor results but in fact has surprisingly good values. We believe that this is due to cyclists illegally using the smooth sidewalk instead of the bumpy road.

Furthermore, cyclists will in practice also avoid the worst potholes and similar bumps if possible, i.e., these will usually be missing from our analysis.

Sensor Inaccuracy

The inaccuracy of the GPS sensors [63] combined with noise produced by the motion sensors of the smartphones form another limitation of the dataset. We tried to partially address these limitations with our preprocessing steps but they can, of course, not be fully mitigated. The only alternative would be using dedicated hardware which, however, will result in significantly less recorded rides due to the adoption barrier.

Temporal Influences

The rides in the SimRa dataset date back up to 2019 and it is possible that the surface quality changed throughout the time. One reason for that could be that the surface type is changed for example from cobblestones to asphalt or potholes and cracks are repaired. This would presumably lead to better results after the change, but show up as two-peak distribution in our dataset. When using this approach in practice, we hence propose to only consider the most recent rides.

Another temporal influence might be the weather. On rainy, windy or snowy days, the surface quality of the road, especially with wet gravel, can suffer significantly, which is not considered by our surface quality analysis approach. However, adding this factor to the analysis, would in our dataset reduce the validity of the results, since it would reduce the number of considered rides.

3.2.5 Related Work of CycleQuality

Surface quality is usually quantified based on the International Roughness Index (IRI), e.g., [76]. Traditionally, this was done using so-called profilographs (approximately resembling a large ladder with wheels) which are towed by a car or pushed manually. A study from the 1980s [26] found that they often break down and are difficult to maneuver in narrow streets which renders them infeasible for measuring the

quality of bike lanes. Furthermore, such measurement are very personnel-intensive which makes it unrealistic to apply them on a broad scale.

Taniguchi et al. [82] detect road hazards such as debris, potholes, or bumps. For this, they attach an ultrasonic distance sensor to a bike handlebar, scanning the ground in front of the bicycle. While the approach can warn cyclists about incoming bumps on the road ahead in real-time, this approach does not scale for city-wide analysis due to the sheer number of sensors needed.

Peng et al. [70] follow the same goal, using motion sensors instead of distance measurements. Data is analyzed offline using a classifier which can identify asphalt, pebbles, and very bumpy underground. In contrast to our work, their emphasis is on identifying specific surface *types* rather than the surface *quality*. Also, their approach again requires dedicated hardware.

Zhou et al. [99] try to detect manhole covers. For this, they analyze a video stream from a bike handlebar-mounted smartphone camera using a convolutional neural network. While their approach is similar to ours regarding hardware, constantly recording video means high power consumption. Furthermore, the phone needs to be mounted in an awkward angle which makes it impractical for every day use.

Luedemann et al. [61] use a smartphone app to record accelerometer data as an indicator for surface quality. Their approach, however, relies on single rides and has no concept of aggregating data from multiple rides.

Similar to us, Yamaguchi et al. [95] want to measure the surface quality. To achieve highly precise results, they combine the smartphone motion sensors with a cyclometer. This approach will always give more precise results than our approach but again requires dedicated hardware which renders a city-wide usage infeasible.

Beyond these, there are several car-centric approaches which either use dedicated hardware, e.g., [22], have very specific phone placement requirements, e.g., [27], or focus on detecting the transients, i.e., individual pot holes, e.g., [2, 57, 58].

3.2.6 Summary of CycleQuality

Cities all over the world aim to increase the modal share of bicycle traffic, e.g., to address emission problems, frequent traffic jams, but also to improve the citizens' health through more daily activity. Aside from safety, a key influence factor for this is comfort, particularly in the form of the surface quality of cycling infrastructure. Monitoring the surface quality manually, however, is infeasible due to the dimensions of such infrastructure.

In this paper, we proposed a crowdsensing approach in which cyclists record their daily rides using a smartphone app and the phone's built-in motion sensors. We proposed a data processing pipeline that starts on the edge (i.e., the phone) and ends in a cloud backend. Furthermore, we showed that our crowdsensing approach can indeed derive surface quality and implemented two use cases for using such data.

3.3 Cyclist Model for SUMO

Active transportation modes such as cycling provide health benefits, alleviate traffic congestion, and reduce air pollution [36]. In practice, however, cyclists often face a car-centric traffic infrastructure which has a significant impact on their (perceived) safety and also affects the attractiveness of cycling routes [3, 50, 69]. Changing this infrastructure to better accommodate cyclists and pedestrians requires significant planning efforts of city planners and traffic engineers. Similarly, road traffic safety systems for vulnerable road users are often assessed using simulation. Particularly the interaction with cars is relevant when it comes to V2X-based safety systems for cyclists [67]. Many of these studies rely on the open source simulation platform SUMO⁹ (Simulation of Urban Mobility), which allows them to study the effects of infrastructure changes before implementing them on the streets.

In SUMO, vehicles and their dynamics are simulated individually [60]. Unfortunately, the cyclist model is not particularly realistic – cyclists can either be modeled to behave as slow cars or as fast pedestrians. Several studies have already improved the bicycle model of SUMO. For instance, Katha et al. [53] investigated the intersection behavior of cyclists using camera traces and transferred findings into SUMO. Also, Grigoropoulos et al. [37] improved modeling of bicycle infrastructure at intersections while Heinovski et al. [42] created a virtual cycling environment to import real bicycle behavior directly into SUMO. Nevertheless, the current cyclist behavior

⁹<https://www.eclipse.org/sumo/>

in SUMO is still rather unrealistic; so far, researchers have devoted much more effort to car models, e.g., [5, 20, 33, 34, 54, 56, 74, 86]. One reason for this is that, until recently, not enough data on real-world cyclist behavior have been available. Today, crowdsourced data collection approaches such as SimRa¹⁰ [50] have made thousands of cycle tracks available as open data.

In this paper, we analyze the SimRa dataset regarding acceleration, deceleration, and velocity of cyclists as well as their left-turn behavior in four-way intersections. We then use our findings to improve the cyclist model in SUMO. Additionally, we add three more detailed cyclist models for slow, medium and fast cyclists. In this regard, we make the following contributions:

- We show that SUMO’s default bicycle simulation is not realistic,
- we improve bicycle simulation of SUMO by deriving new parameters for that vehicle type in SUMO,
- we add three new bicycle simulation models - slow, medium and fast - to SUMO by splitting the SimRa dataset into slow, medium and fast rides.
- we develop an intersection model which captures cyclists’ left-turn behavior at intersections in a more realistic way, and
- we compare our improvements to SUMO’s default bicycle simulation, using the SimRa dataset as a ground truth.

¹⁰<https://github.com/simra-project/>

This paper is an extension of our previous work [51]. The main new contributions are:

- We rerun all analyses of the original paper using significantly more rides which have become available in the time period since starting our original paper.
- We complement the general cyclist model from our original paper with three dedicated models covering the behavior of cyclist groups clustered by their average velocity. For each of the three models, we also derive distributions of left-turn, acceleration, and maximum velocity behavior.
- For both the general and the three dedicated cyclist models, we also derive deceleration behavior as a distribution.
- We implement all new contributions as SUMO plugins and have successfully reimplemented the cyclist model from our original paper in SUMO core. We are currently in the process of finalizing an implementation of the extensions of this paper also in SUMO core.
- We evaluate all four cyclist models against the default cyclist model of SUMO using scenarios from Berlin, Munich, and Hanover, Germany

3.3.1 Cycling Behavior in SimRa and SUMO

In this section, we analyze real-world cyclists’ behavior extracted from the SimRa dataset and compare it to the behavior of SUMO’s default bicycle model. SimRa’s dataset stems from crowdsourced smartphone data generation and thus suffers from poor sensor quality [23, 90] as well as heterogeneous hardware and users [6]. This leads to a lot of unclean data, which we first need to filter out (??). Since we want to create one general type for all cyclists and complement it with three dedicated models for slow, medium, and fast cyclists, we need to derive three distinct types from the SimRa dataset, which we do in ???. We then analyze acceleration, deceleration and velocity behavior for each of the four models in ?????? before discussing left-turn behavior at four-way intersections of the different models in ???. We omit a detailed discussion of the right-turn behavior at intersections, since SUMO’s default model does not deviate much from the behavior observed from SimRa’s dataset. When referring to SUMO’s bicycle model, we refer to the “slow car” model of SUMO as the “fast pedestrian” model occasionally leads to poor results and was therefore not considered further.

Aside from the public SimRa datasets [10, 11, 49] and more recent rides available on GitHub,¹¹ we also used non-public rides which have, for privacy reasons, not been published yet. Table 3.5 summarizes the most important attributes of the dataset that we used.

¹¹<https://github.com/simra-project/dataset>

Table 3.5: Most important attributes of entities of the SimRa dataset

	Total	Used
Rides	60 470	55 175
Accelerations	12 382 675	2 330 292
Decelerations	12 985 955	2 623 904

We used rides from almost 100 SimRa regions when calculating the distributions for the maximum acceleration, maximum deceleration and maximum velocity of cyclists.

Preprocessing

To achieve the best possible data quality, we tested various pre-processing techniques and filters. We also conducted an experiment in which sample trajectories were recorded in parallel on several SimRa client devices and compared to a ground truth trajectory recorded by a stand-alone GPS receiver. In the end, we used a Gaussian Kernel filter for improving location data and a Low Pass filter for the velocity data. Additionally, the SimRa dataset contains information about the location accuracy, which we use to filter out rides where the GPS accuracy suffered greatly. After filtering semantically and syntactically defective files, we used data from 55 175 rides, which is around 65% of the initial dataset as input for our analysis scripts.¹²

¹²<https://github.com/simra-project/SimRaXSUMO>

Categorizing Cyclists by Velocity

To get different types of cyclists, we decided to analyze the average velocity of each ride after filtering out the stops. Figure 3.18 shows the distribution of the average velocities of each ride in the SimRa dataset. This does not present an obvious way to split the dataset into three types, which is why we have split the dataset from a SUMO users' perspective. We, hence, decided to split the dataset so, that the 25% slowest and the 25% fastest rides represent the slow and the fast cyclists respectively, which leaves the middle 50% to the medium-paced cyclists. This results in the following cyclist type velocities: *Slow cyclists* have an average cycling velocity of up to 13.5 km/h. *Medium cyclists* have an average cycling velocity between 13.5 km/h and 17.9 km/h. *Fast cyclists* have an average cycling velocity above 17.9 km/h. We have chosen against an equal split and in favor of a 25%-50%-25% split, because we think that the majority of the cyclists should be in the same cyclist type, namely, the medium cyclist type.

Acceleration

For analyzing cyclist acceleration, we extracted acceleration maneuvers from the dataset. For this, we slightly adapted the approach of [62]. First, we split the velocity profile at its local extrema to get segments where the cyclist accelerates/decelerates. Then, we consider only segments with a distance from 20 m to 350 m and a duration between 5 s to 40 s. We also make sure to filter out segments where the variance in

velocity is too low, i.e., such that $\frac{|v_s - v_e|}{\max(v_s, v_e)} > 0.5$, where v_s and v_e are the velocities at the start and end of a segment. With that approach we found 228 347 acceleration maneuvers in the cleaned dataset. Distribution fitting processes, which were done with SciPy¹³, showed that the Burr (Type XII) distribution [18] $Burr12(x; c, d) = c * d * \frac{x^{c-1}}{(1+x^c)^{d+1}}$ for $x = a_{max} \geq 0$ and $c, d > 0$ fits the data best for the general cyclist model (see also fig. 3.19a). For the models of the slow and fast cyclist models, the Burr (Type III) distribution [18]

$Burr3(x; c, d) = c * d * \frac{x^{-c-1}}{(1+x^c)^{d+1}}$ for $x = a_{max} \geq 0$ and $c, d > 0$ fits the data best, while the Mielke Beta-Kappa distribution [64]

$M\beta\kappa(x; c, d) = \frac{k*x^{k-1}}{(1+x^s)^{1+k/s}}$ for $x = a_{max} > 0$ and $k, s > 0$ is the best fit for the medium cyclist model's acceleration distribution (see also fig. 3.19b).

Comparing the acceleration capability of actual cyclists (the SimRa dataset) with the default SUMO cyclist model, differences become apparent. By default, SUMO specifies a_{max}^{SUMO} with 1.2 m/s^2 . This deviates significantly from the findings in the SimRa dataset where only 15% of the acceleration maneuvers are executed with a maximum acceleration of 1.2 m/s^2 or higher. Furthermore, the empirical distributions are rather wide, indicating a broad variance across different cyclist types and cycling situations, which is in stark contrast to SUMO's strategy of choosing a fixed maximum value.

¹³<https://scipy.org/>

Deceleration

For analyzing cyclist deceleration, we extracted deceleration maneuvers from the dataset (see ??). Distribution fitting processes showed here that the Johnson's S_U -distribution [47] $JSU(x; a, b) = \frac{b}{\sqrt{x^2+1}}\phi(a + b * \log(x + \sqrt{x^2 + 1}))$ for $x = d_{max}$ and $b > 0$ with ϕ being the probability density function of the normal distribution, fits the data best for the general cyclist model (see also fig. 3.20a). $JSU(d_{max}; a, b)$ was also the best fit for the slow and medium cyclist models, whereas the fast cyclists' data was the best fit for the Student's t -distribution [80] $t(x; \nu) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\pi*\nu*\Gamma(\nu/2)}}*(1+x^2/\nu)^{-(\nu+1)/2}$ for $x = d_{max}$ (see also fig. 3.20b).

Comparing the deceleration capability of actual cyclists (the SimRa dataset) with the default SUMO bicycle model, differences become apparent. By default, SUMO specifies d_{max}^{SUMO} with -3 m/s^2 . This deviates extremely from the findings in the SimRa dataset where none of the deceleration maneuvers are executed with a maximum acceleration of -3 m/s^2 or lower. Furthermore, the empirical distributions are rather wide, indicating a broad variance across different cyclist types and cycling situations, which is in stark contrast to SUMO's strategy of choosing a fixed maximum value.

Velocity

To gain insights into cyclists' behavior regarding their velocities, we calculate the maximum velocity for each ride file in the cleaned SimRa dataset. Using distri-

bution fitting, we found that the symmetric generalized normal distribution [65] $SGND(x; \beta) = \frac{\beta}{2*\Gamma(1/\beta)} * exp(-|x|^\beta)$ for $x = v_{max}$ fits the empirical data of all cyclists best (see also fig. 3.21a) and is therefore a valid fit for the specification of the empirical distribution of v_{max}^{SimRa} for the general cyclist model. For the model of the slow cyclists, the Johnson's S_U -distribution [47] $JSU(v_{max}; \Phi^*)$ fits the data best, while the best fit for the medium cyclists' model is the Exponentially Modified Gaussian distribution [39] $EMG(x; K) = \frac{1}{2*K} * exp(\frac{1}{2*K^2} - x/K)erfc(-\frac{x-1/K}{\sqrt{2}})$ for $x = v_{max}$ and $K > 0$. Finally, the non-central t -distribution [44] $NCT(v_{max}; \Phi^*)$ emerged as the best fit for the fast cyclists' data.

On the other hand, SUMO sets v_{max}^{SUMO} at 5.56 m/s by default. This deviates significantly from the findings in the SimRa dataset where 77% of the rides have a higher maximum velocity. Bringing this together with the acceleration findings, real-world cyclists often (but not always) cycle much faster than SUMO cyclists and vary much more in their acceleration behavior.

Left-turn Behavior at Intersections

According to the SimRa dataset, cyclists either behave like cars (using the normal road) or pedestrians (using the pedestrian crossing) to take left-turns at intersections. We call the former a *direct* left turn and the latter an *indirect* left turn.

SUMO's default model only provides cyclists with an unrealistic "bicycle-lane-to-bicycle-lane-left-turn" (see also fig. 3.22), where the cyclist enters the intersection

from a bicycle lane, crossing all car lanes and directly entering the bicycle lane again.

Taking a closer look at real world intersections in the SimRa dataset revealed that there are mainly two intersection types. In the first intersection type, the *indirect* path is chosen with a probability of 61% when all cyclists are considered, while medium cyclists and fast cyclists prefer the indirect path with a probability of 87% and 50% respectively. However, on the second intersection type, almost all cyclists choose the *indirect* path. Slow cyclists almost never tend to do direct left-turns, since they presumably avoid car traffic the most. Randomly investigating intersections of both types revealed that the first intersection type has no specific characteristics while the second intersection type actively encourages cyclists to indirect turns through the design of the intersection, e.g., by having a traffic island in the center. Since such information cannot be identified in OpenStreetMap ([OSM](#)) data reliably and in an abstract way, we will consider only the first intersection type in the following.

SUMO and real world data differ decisively in all metrics considered, namely acceleration, velocity, and left-turn behavior at intersections. However, these three metrics are crucial for realistically simulating bicycle traffic. In the following, we try to adapt SUMO to simulate a more realistic cyclist behavior by introducing three different cyclist types.

3.3.2 Improving SUMO’s Bicycle Simulation

To improve the simulation, we propose three changes to SUMO’s bicycle model: First, the longitudinal kinematic parameters of SUMO’s default bicycle model are (re-)parameterized based on the findings from the SimRa dataset. Second, a novel simulation model is derived from SimRa trajectories to exclusively simulate realistic left-turn bicycle behavior at intersections based on the findings in ???. The latter model is referred to as the intersection model in the following. Third, four different cyclist models, with different longitudinal kinematic and left-turn behaviors depending on the findings from ???. One model is for all cyclists combined and the three other models are for modeling slow, medium and fast cyclists.

Longitudinal Kinematic Behavior

In ??, we derived maximum acceleration, maximum deceleration and maximum velocity characteristics from the SimRa dataset. We now use them to improve the longitudinal kinematic behavior of the default SUMO bicycle model. Contrary to the default parameterization, we use theoretical distribution functions instead of scalar values for the exposed kinematic parameters. This enables the model to produce more realistic bicycle simulation results since the heterogeneity of real world cycling styles is reflected.

We derive the theoretical distributions by aggregating the respective features from ??????. For this, we rely on the *law of large numbers* which states that the average of the results obtained from a large number of trials of the same experiment eventually converges to its true expected value [30]. In the context of this work, this means that individual rides do not matter but that the aggregates of multiple rides will converge towards their actual expected value given a sufficiently large number of rides.

For the implementation, we used *vTypeDistributions* following the results of our previous analysis and sample both distribution independently.

It should be noted that through the parameterizations with theoretical probability density functions SUMO's *speedDev* parameter becomes obsolete as variance between the kinematic preferences among cyclists are already represented by the distribution function.

Furthermore, alternatives for the acceleration parameters have been added to SUMO. This would enable a user to pick a normal distribution and choose the parameters for it. This would yield more realistic scenario data contrary to using a simple scalar value for the acceleration.

Left-turn Behavior at Intersections

To improve the degree of realism in cyclists' left-turn behavior at signaled intersections, we use an adapted version of the external intersection model (a Python script

that steers cyclists via SUMO’s *Traffic Control Interface*) as proposed by Kathis and Grigoropoulos [53] which is based on previously recorded real-world trajectories as their guidelines for cyclists across a single predefined intersection. Our approach algorithmically synthesizes the cyclists’ trajectories (i.e., their respective guidelines across the intersection) for any regular four-way intersection and can therefore be seen as a step towards a more universal solution.

The left-turn maneuver distribution, as we call it, specifies the probability of the cyclists choosing either the *direct* or the *indirect* path to cross the intersection. For this, we use the distributions derived in ?? as the default for our intersection model for each different cyclist group. Users, however, can adjust the distributions if desired or needed for their specific purposes (see also the exception cases in ??).

In order to simplify the process and to improve performance, we decided to integrate this feature into the SUMO core. The implementation provides a new parameter that lets the user adjust the indirect left turn probability of a bicycle ride. To detail the workflow, the cyclist makes a decision before each intersection, where there is at least one direct and indirect left-turn available to choose from. This delegates decisions inside SUMO just by using the bicycle vehicle type parameter.

Different Cyclist Models

Our approach defines distributions of models derived from the same default implementation of SUMO. The models vary only in the parameter values used. The parame-

ters considered are maximum acceleration (*accel*), maximum deceleration (*decel*) and maximum velocity (*maxSpeed*), splitted into 3 groups, as it can be seen in [??????](#). In order to leverage distribution parameters for a bicycle model, we used the *vType-Distribution* from SUMO to define a distribution of cyclists having all the parameters mentioned.

With this data ready, the only remaining step is to augment these groups with a left-turn behavior probability per group. In order to make a decision whether to do a direct or an indirect left-turn, we approached the problem by using a script that controls the cyclists, as described in [??](#).

Thus, this group-based model enables the user to simulate bicycle traffic in SUMO more realistically.

3.3.3 Evaluation of the Cyclist Model for SUMO

In this section, we evaluate the new cyclist models from [??](#) by comparing them to each other and to SUMO’s default simulation model and the real-world data taken from the SimRa data set. We start by introducing the simulation setup ([??](#)) which we used to create and run the scenarios with. We then continue analyzing acceleration ([??](#)), deceleration ([??](#)), velocity ([??](#)), and left-turn behavior at intersections ([??](#)) before evaluating the combination of all model extensions ([??](#)). Please note: While it may appear obvious that using the SimRa data set for both parameterization and

evaluation should lead to perfect results, this is not the case as our extensions are subject to the design restrictions imposed by SUMO

Simulation Setup

As SUMO users can import real-world scenarios from OSM data, simulation results can be compared to real-world data and thus be evaluated. For our evaluation, we chose specific traffic scenarios from multiple SimRa regions that are representative and likely to showcase both strengths and weaknesses of our extensions. Likewise, we do not show every cyclist type's results in detail to avoid too many figures and tables. For evaluating the longitudinal behavior (acceleration, deceleration and velocity), we chose urban traffic scenarios with long straight sections. As example locations, we use *Oranienstraße* in Berlin, *Dachauer Straße* in Munich and *Frauentorgraben* in Nuremberg. For evaluating left-turn behavior, we chose compact scenarios around signaled intersections with multiple lanes on each axis. For this, we study three intersections in Berlin, namely at *Mehringdamm* (see also fig. 3.23), *Warschauer Straße*, and *Alexanderstraße*.

We made our scenarios as realistic as possible by choosing main roads and intersections with a lot of traffic and also added a significant number of cars, as table 3.6 shows.

For our evaluation, we use SUMO version 1.14.0 and a step size of 1 s in simulations. The SUMO default results are obtained with SUMO's *vType Bicycle* for cyclists,

Table 3.6: Most important attributes of our simulation scenarios.

	Cars	Bicycles	Distance	Lanes
Oranienvstr.	3158	300	1528	—
Dzchauer Str.	3427	300	1204	—
Frauentorgraben	1625	300	1023	—
Mehringdamm	180	2700	—	29
Warschauer Str.	180	2700	—	26
Alexanderstr.	180	2700	—	25

i.e., the maximum acceleration, maximum deceleration and maximum velocity are scalars and set to 1.2 m/s^2 , -3 m/s^2 and 5.56 m/s respectively.

Acceleration

fig. 3.24a shows the empirical distributions of the maximum acceleration among all cyclist types inside the *Frauntorgraben* scenario simulation and the corresponding real world data. It is evident that real-world acceleration maneuvers show heterogeneous maximum rates of acceleration. The same is true for the three cyclist groups of our new approach as fig. 3.24b shows. Note that the other two cyclist groups, namely slow and medium, allow the same conclusion and, that we have randomly chosen the slow cyclist group to visualize in fig. 3.24b to make the figure more readable. Apparently, the default parameterization is not suitable to describe this acceleration behavior among cyclists, as it provides homogeneous maximum acceleration rates within the simulation. Our new parameterization - for all cyclists together and for each of the cyclist types - is significantly closer to the real-world behavior in the SimRa data set with its highly heterogeneous behavior across cyclists. This can also be seen, when comparing the mean, standard deviation and median values of SUMO default, all and slow (*s*) cyclists (both in the SimRa dataset and our approach) depicted in table 3.7. It also shows, that the division of the cyclist into subgroups further increases the realism, since the difference between the mean and median values of our approach (*all*) and Simra (*all*) shrinks, compared to our approach (*s*) and Simra (*s*). The standard deviation suffers though, which is probably due to less rides being

Table 3.7: Results Overview Maximum Acceleration Comparing SUMO Default, Our Approach (All), Our Approach (Slow), SimRa (All) and SimRa (Slow)

	Mean	Std. Deviation	Median
SUMO default	1.178	0.019	1.18
Our approach (<i>all</i>)	0.921	0.218	0.95
SimRa (<i>all</i>)	0.761	0.217	0.71
Our approach (<i>s</i>)	0.727	0.168	0.74
SimRa (<i>s</i>)	0.656	0.109	0.65

regarded, because of the splitting into three subgroups. Please note also here, that we omitted the medium and fast cycling groups to stay consistent with fig. 3.24b and avoid clutter. That our new parameterizations are not a perfect fit indicates that there are probably additional influence factors, e.g., the traffic density or the weather situation, not covered in our kinematic models which aggregate data from all SimRa rides.

Deceleration

fig. 3.25a shows the empirical distributions of the maximum deceleration among all cyclist types inside the *Dachauer Straße* scenario simulation and the corresponding real world data. Here, we only show the medium cyclist group as an example to avoid

Table 3.8: Results Overview Maximum Deceleration Comparing SUMO Default, Our Approach (All), Our Approach (Medium), SimRa (All) and SimRa (Medium)

	Mean	Std. Deviation	Median
SUMO default	2.664	0.507	2.77
Our approach (<i>all</i>)	1.071	0.336	1.09
SimRa (<i>all</i>)	0.962	0.141	0.96
Our approach (<i>m</i>)	0.949	0.362	0.93
SimRa (<i>m</i>)	0.973	0.100	0.98

clutter, the other two cycling groups (slow and fast) show very similar results. It is evident that real-world deceleration maneuvers show heterogeneous maximum rates of deceleration. The same is true for the medium cyclists as fig. 3.25b shows. Here, too, the default parameterization is not suitable to describe this deceleration behavior among cyclists, as it provides homogeneous maximum deceleration rates within the simulation. Our new parameterization - for all cyclists together and for each of the cyclist types - is significantly closer to the real-world behavior in the SimRa data set with its highly heterogeneous behavior across cyclists. Just like with the maximum acceleration, table 3.8 shows, that our approach is not only more realistic than SUMO's default bicycle model (both with all and medium (*m*) cyclists), but the introduction of the cyclist groups increases the realism. The medium cyclist group was randomly chosen as a representative for the other groups, since the conclusion

does not differ. Influence factors, such as the traffic density or the weather situation, not covered in our kinematic models which aggregate data from all SimRa rides, result in a non-perfect fit for our new parameterizations. Note, that we had to filter out deceleration values, that we deemed too high (above 7 m/s^2), to avoid recording emergency deceleration, which is another parameter in SUMO's vehicle parameterization.

Velocity

fig. 3.26a shows the empirical distributions of all cyclists' maximum velocities in the *Oranienstraße* scenario simulation and the real-world scenario. As with maximum acceleration and deceleration rates, maximum velocities vary widely among real-world cyclists and we only depict one cyclist group (this time fast cyclists) as an example for other groups to increase the readability of the figure. Once more, the default parameterization is not able to reflect this characteristic. This is also the case for the different cyclist groups as fig. 3.26b shows exemplary. According to table 3.9 the maximum velocity aspect of our simulation is the closest, when compared to maximum acceleration and maximum deceleration of all and fast (*f*) cyclist groups in SimRa and our approach. This is probably due to the fact, that we splitted the cyclist based on the average velocity. Our new parameterization is thus significantly closer to the real-world behavior of cyclists. As for acceleration and deceleration behaviors, the fact that our new parameterization is not a perfect fit to the real-world

Table 3.9: Results Overview Maximum Velocity Comparing SUMO Default, Our Approach (All), Our Approach (Fast), SimRa (All) and SimRa (Fast)

	Mean	Std. Deviation	Median
SUMO default	5.555	0.002	5.556
Our approach (<i>all</i>)	6.589	1.234	6.58
SimRa (<i>all</i>)	7.072	0.904	7.10
Our approach (<i>f</i>)	8.272	0.902	8.20
SimRa (<i>f</i>)	7.872	0.738	7.86

data indicates that there are likely to be additional influence factors not captured in our model.

Left-turn Behavior at Intersections

As shown in fig. 3.27, which shows the intersection between Alexanderstraße and Karl-Marx-Allee in Berlin, the 2D trajectories produced by the new intersection models converge towards the trajectories of the SimRa data set. While the trajectories produced by SUMO’s default bicycle model, as can be seen in fig. 3.22, only offer direct ”bike-lane-to-bike-lane” turns, the new model is significantly closer to real-world intersection behavior of cyclists. This is true for all of our cyclist models. Here, we see again the different left-turn behaviors of different cyclist types. While

slow cyclist only prefer the indirect left-turn, the medium and fast cyclists are more inclined to take the direct left-turn comparatively.

Combining Intersection Model and Kinematic Extension

To achieve a holistic comparison between SUMO’s default bicycle model and our new models, we measure the durations of left-turn maneuvers at multiple intersections and compare the empirical distributions of these measurements. To specifically monitor the impact of our changes, we do not include any ride time before or after the intersection in the measurements. Note that we also omit the slow rides, since no or too few slow rides went through the intersections presented here.

Based on this, we identified the following four findings: First, our new models outperform the default at most intersections, as its measured durations converge with real data, see for example fig. 3.28. Especially when given the option to use the *indirect* path, cyclists take longer to cross an intersection as they need to stop at an additional traffic light. This is consistent with real-world data as we find it in the SimRa dataset at multiple intersections.

Second, in some cases, we were able to improve our results by adjusting the left-turn behavior distribution following the second distribution discussed in ???. The “lane only” results in fig. 3.29 were achieved by prohibiting cyclists from using the *direct* path. Obviously, it takes much longer for cyclists to cross the intersection than SUMO’s default simulation model suggests. When examining SimRa trajectories at

this particular intersection, almost all cyclists chose the *indirect* path as the infrastructure guides cyclists to do so. Hence, manually adjusting the left-turn behavior distribution for such intersections is crucial.

Third, at a few intersections, the results of our approach do not yet sufficiently reflect real-world bicycle behavior (see fig. 3.30). We discuss possible reasons for this in ??.

Fourth, creating three cyclist models based on their average velocity also revealed, that SUMO’s default model behaves like fast cyclists. This becomes clear when looking at figs. 3.28 to 3.30, since the lines representing the cyclists modeled with our fast cyclist model are the closest to the red line representing SUMO default.

3.3.4 Discussion of the Cyclist Model for SUMO

Overall, the results presented in this paper show a significant improvement over the state-of-the-art. Nevertheless, they still have a number of shortcomings. In this section, we discuss the inherent limitations of our approach in general (??) as well as problems resulting from the SimRa dataset as our ground truth data (??). We also describe a potential problem regarding e-bikes in ??.

Methodological Challenges

Generalization Problem Our initial assumption was that the behavior of cyclists in a single intersection cannot be generalized to all intersections [53] but that the average

behavior across a large number of intersections will be close enough to cyclists' behavior at arbitrary intersections. This seems to be true only for a (relatively large) subset of intersections – apparently, the intersection behavior of cyclists is more heterogeneous than expected. We believe that this is due to the fact that we averaged across all intersections in our dataset whereas there are apparently different classes of intersections that we did not account for.

Primarily, the intersection design is likely to have a strong impact: Consider the example in fig. 3.31 where a traffic island partially blocks direct left turns and where markings on the ground suggest indirect turns. As another example, the intersection Bismarckstraße/Leibnizstraße had no direct left turns in the SimRa dataset. In this intersection, the reason would be that cyclists legally have to use a bike lane. When using that bike lane, a direct left turn would require cyclists to first pass through a row of parked cars, then to cross four car lanes of a major street before being able to turn left.

Aside from that, other possible influence factors include the amount and velocity of traffic (higher numbers of cars or faster cars can be expected to lead to more indirect turns), gender and age group distributions of cyclists in the respective intersections, as well as weather and light conditions or the grade of the street. In future work, we plan to explore these possible influence factors, focusing on the intersection design which we deem to have the strongest impact.

Another problem results from inaccuracy in the dataset used: GPS and motion sensors of smartphones provide only imprecise insights into actual “micro”-behavior of cyclists. Using a broad group of cyclists as input will always result in this limitation which we tried to overcome based on preprocessing and filtering of the SimRa dataset. Alternatives would be additional sensors (especially cameras) on bicycles or on intersections as in [53]. These, however, have the inherent limitation that they will either limit the number of bicycles producing data or the number of intersections covered.

Dataset Choice as Ground Truth

In this paper, we used the SimRa datasets as input for our analysis as it is, to the best of our knowledge, the first public dataset comprising a large number of rides that actually publishes individual rides in an anonymized but non-aggregated form. We need to keep in mind, however, that SimRa was designed for a different purpose: For example, the SimRa app records motion sensors at 50 Hz but only persists every fifth value of a moving average over 30 values. While this suffices for detecting near miss incidents [50], it further limits the resolution of motion data (and thus any conclusions we can draw from that). Furthermore, the SimRa data which we used were recorded over a period of 1.5 years. During such a long period of time, physical changes to the bicycle infrastructure (both temporary and permanent) will occur, thus, adding additional noise to the data.

Finally, SimRa relies on crowdsourcing as a data collection method which often leads to participation inequality. As a result, individual users will be overrepresented in some intersections and street segments and not represented in others. Furthermore, based on the data collection method using smartphones, the user group of SimRa is likely to have a slight gender bias towards males and an age group bias towards cyclists between the ages 20 and 50. These biases will, of course, be reflected in our analysis results and cannot be compensated unless other cycling datasets become available in non-aggregated form.

3.3.5 Generalizability

The SimRa dataset which we used to analyse the longitudinal kinematic behavior contains rides from almost 100 regions from Germany, Switzerland and Austria, which increases the generalizability. However, to derive and evaluate the left-turn model, we need data from intersections with many left-turns, with the same orientation, e.g., entering the intersection from the south and leaving it to the west. Mind, that we also split the rides into three groups: slow, medium and fast. Only the regions Berlin, Munich and Nuremberg provide eligible data for this, since almost half the rides are from these regions [52]. This is further aggravated by the fact that the preprocessing step (see ??) further reduces the number of eligible rides and with that the number of intersections with sufficient left-turn maneuvers. Nevertheless, this does not pose a threat to the generalizability of our findings – at least for Germany

– as traffic infrastructure guidelines throughout Germany are standardized. Furthermore, adjacent countries such as Austria often also have comparable infrastructure. Additionally, this problem is not present with the evaluation of the longitudinal kinematic behavior, since we only need more or less straight routes without any sharp turns. We hence believe that our findings also apply to at least these countries.

E-bikes

Another potential threat to the realism of our approach are e-bikes, since they support the cyclist, potentially leading to higher acceleration and velocity. To analyze this, we filtered out the e-bikes and found, that the SimRa dataset only contains about 4000 e-bike rides, which is less than 5%. We redid the analyses of the three cyclist groups and found out, that the change in the results is barely noticeable. And there are not enough e-bike rides to create a dedicated e-bike model with a high credibility.

3.3.6 Related Work of Bicycle Simulation for SUMO

In this section, we give an overview of related work on improving intersection behavior (??) and longitudinal (??) behavior of cyclists in SUMO’s simulation models.

Intersection Behavior of Cyclists

Kaths and Grigoropoulos [53] aim to address the shortcomings of SUMO’s intersection model for cyclists. For this, they record video footage of an example intersection in Munich and derive cyclist trajectories. From the set of trajectories, they select one representative trajectory for each combination of start and end points in the intersection and make it available to SUMO via an external API. While this is a significant improvement in realism over SUMO’s intersection model, it is hard to generalize to other intersections and cannot cover the plurality of trajectories chosen by real-world cyclists.

Similar to Kath and Grigoropoulos [53], Grigoropoulos et al. [38] analyze video footage of intersections with the goal of better understanding the intersection behavior of cyclists. Their focus, however, is not on deriving an improved intersection model but rather on identifying best practices for traffic planners working on intersections with high volumes of cycling traffic. Grigoropoulos et al. [37] propose to adjust the default traffic infrastructure inside SUMO to achieve more realistic cyclist behavior at intersections. Here, they focus on the number and shape of bicycle lanes which, however, are highly specific and differ from intersection to intersection.

Longitudinal Behavior of Cyclists

Twaddle and Grigoropoulos [88] examine four models for the longitudinal kinematic behavior of cyclists, i.e., acceleration and velocity. The first, called Constant Model, is the simplest and is the SUMO default: Cyclists accelerate and decelerate at a constant rate until the desired velocity is reached. This model works well when breaking to a full stop but leads to frequent acceleration jumps between a fixed positive or negative value and zero, which is not realistic cyclist behavior. In the Linear Decreasing Model, maximum acceleration is reached when starting the acceleration maneuver and then linearly declines until the desired velocity is reached. This model is outperformed by all other models. In the third and fourth models, Polynomial and Two Term Sinusoidal Model, acceleration or deceleration start at zero and then gradually grow over time. In their paper, Twaddle and Grigoropoulos [88] analyze the video recordings of 1,030 rides in four intersections in Munich, Germany and conclude that the Polynomial Model has overall the most realistic cyclist behavior but is, however, not trivial to implement in SUMO.

A different approach of achieving realistic cycling behavior in SUMO is taken by Heinovski et al. [42]. The authors simulate multiple traffic scenarios in which accidents between cars and cyclists occur to investigate the effects of wireless communication between cyclists and other road users in the context of accident prevention. In order to obtain realistic cycling behavior for SUMO, they set up a novel Virtual Cycling Environment (VCE) featuring an actual bicycle that is connected to the

simulation via multiple sensors. The VCE supports interactive empirical studies in a physically safe environment and allows the authors to record the cyclists' behavior in the form of trajectories. They use a set of recorded trajectories from different cyclists for emulating realistic cycling behavior inside SUMO to simulate accidents. Although their approach produces trajectories from cyclists created with an actual bicycle, it does only achieve limited realism, since no other road users were present when recording the trajectories. Furthermore, deriving a realistic set of trajectories requires a large number of test persons.

3.3.7 Summary of Bicycle Simulation for SUMO

Increasing the modal share of cyclists to provide health benefits, alleviate traffic congestion, and reduce air pollution requires significant planning efforts of city planners and traffic engineers towards an improved cycling infrastructure. For this, city planners often rely on the open source simulation platform SUMO to study the effects of infrastructure changes before implementing them on the streets. Likewise, research on V2X-based safety systems for cyclists often relies on SUMO for evaluation. Unfortunately, SUMO cyclists are either modeled as slow cars or as fast pedestrians, neither of which is overly realistic.

In this paper, we used the recently published SimRa dataset, which to our knowledge is the first public dataset providing detailed insights into a large number of individual cyclists' rides, to improve SUMO's cyclist model. For this, we split the rides into

three categories based on their average velocity, after the stops are filtered out: slow, medium, and fast. We then derived acceleration, deceleration and velocity behaviors for each of these three cyclist groups and reparameterized the SUMO cyclist models. As a SUMO extension, we also developed a new intersection model describing left-turn behaviors of cyclists of the three new groups in four-way intersections. While our work significantly improved the existing cyclist model, it is not as realistic as we wanted it to be. We, hence, discussed a number of research directions which we plan to explore in the near future.



(a) Straßburger Straße



(b) Goethestraße

Figure 3.17: Straßburger Straße and Goethestraße in Berlin are paved with large cobblestones and the roads are contested by cars that are parking or searching for a parking spot. In contrast, the sidewalks are paved with flat-surfaced paving stones and are quiet due to the absence of shops, restaurants or cafes. (Source: Apple Maps)

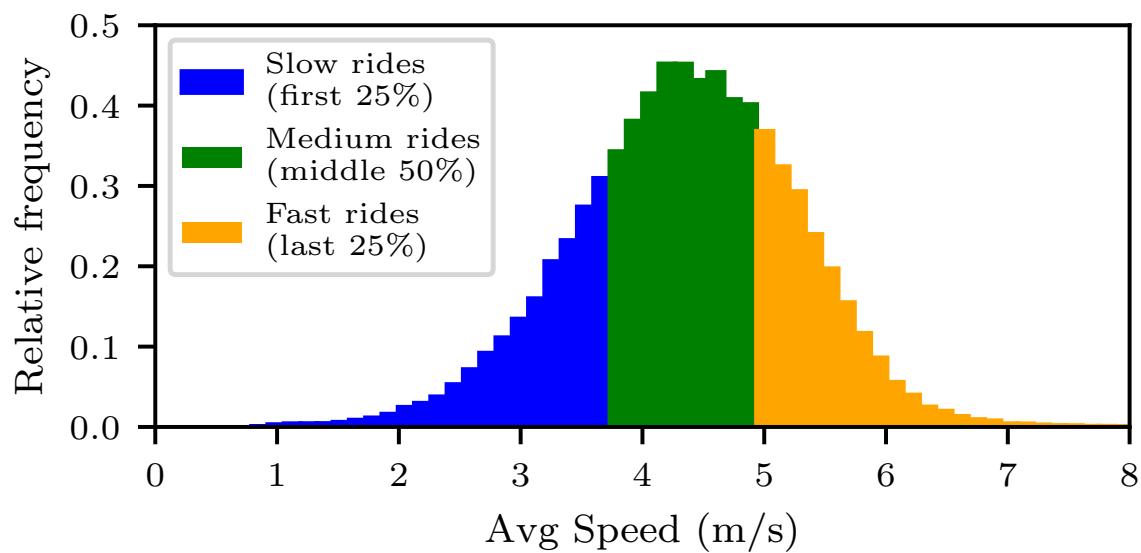


Figure 3.18: Histogram of the empirical average velocity capabilities of cyclists found in the SimRa dataset. The average velocity of all bicycle rides after the preprocessing is 4.38 m/s with a standard deviation of 0.89 and a median of 4.42 m/s.

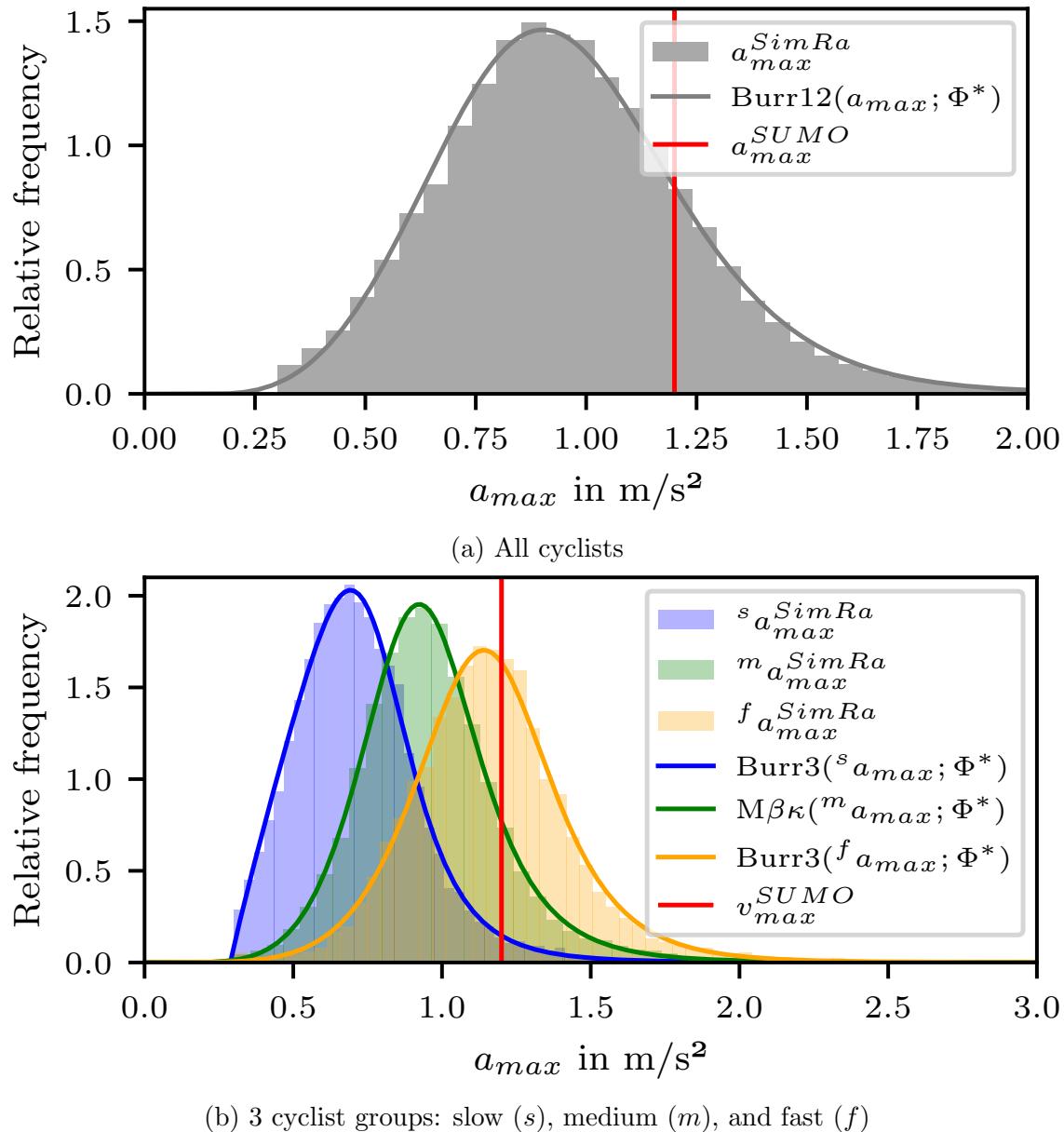


Figure 3.19: Histogram of maximum acceleration capabilities found in the empirical SimRa dataset and their respective distribution functions. The red scalar represents the default value in SUMO.

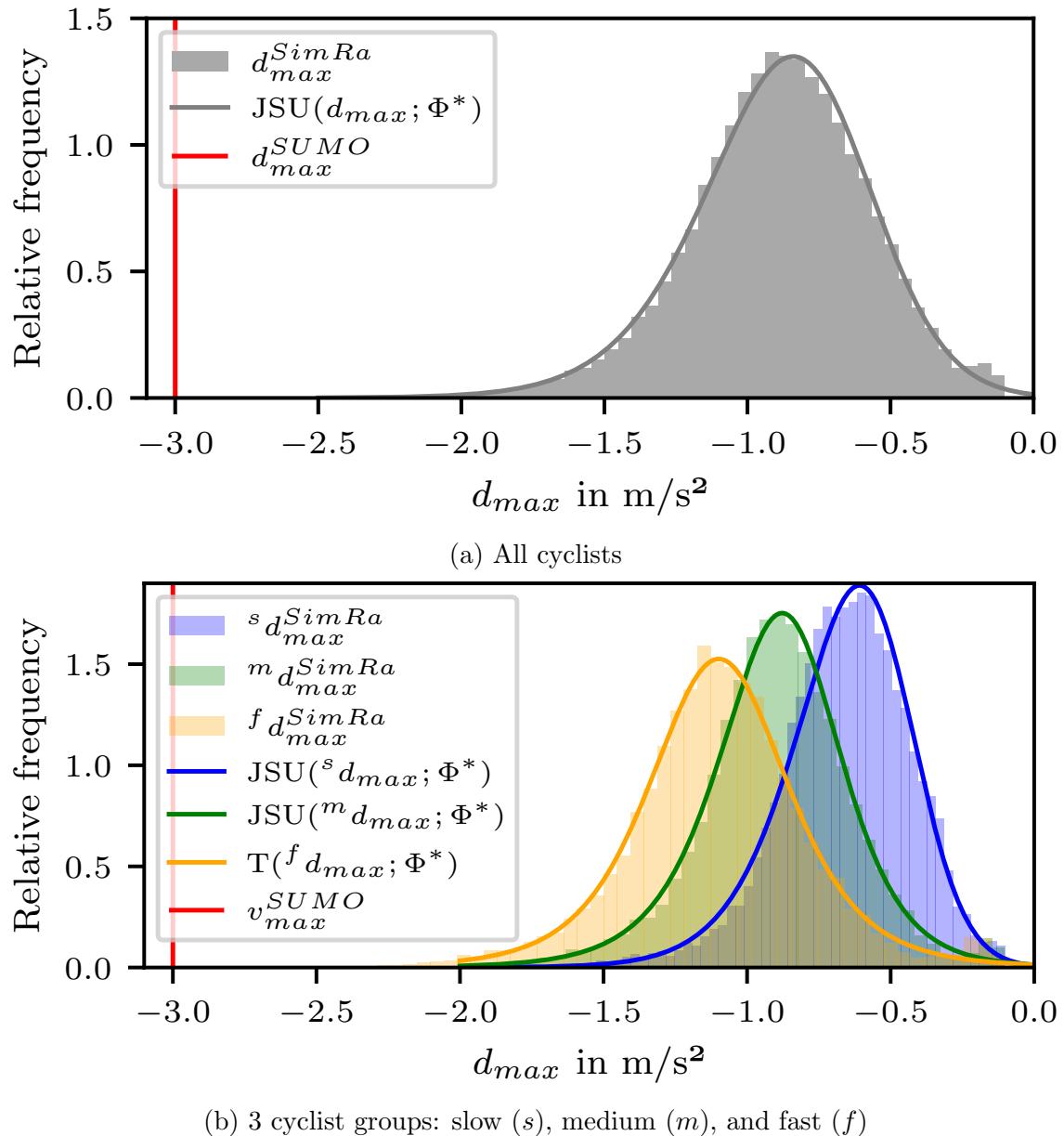


Figure 3.20: Histogram of maximum deceleration capabilities found in the empirical SimRa dataset and their respective distribution functions. The red scalar represents the default value in SUMO.

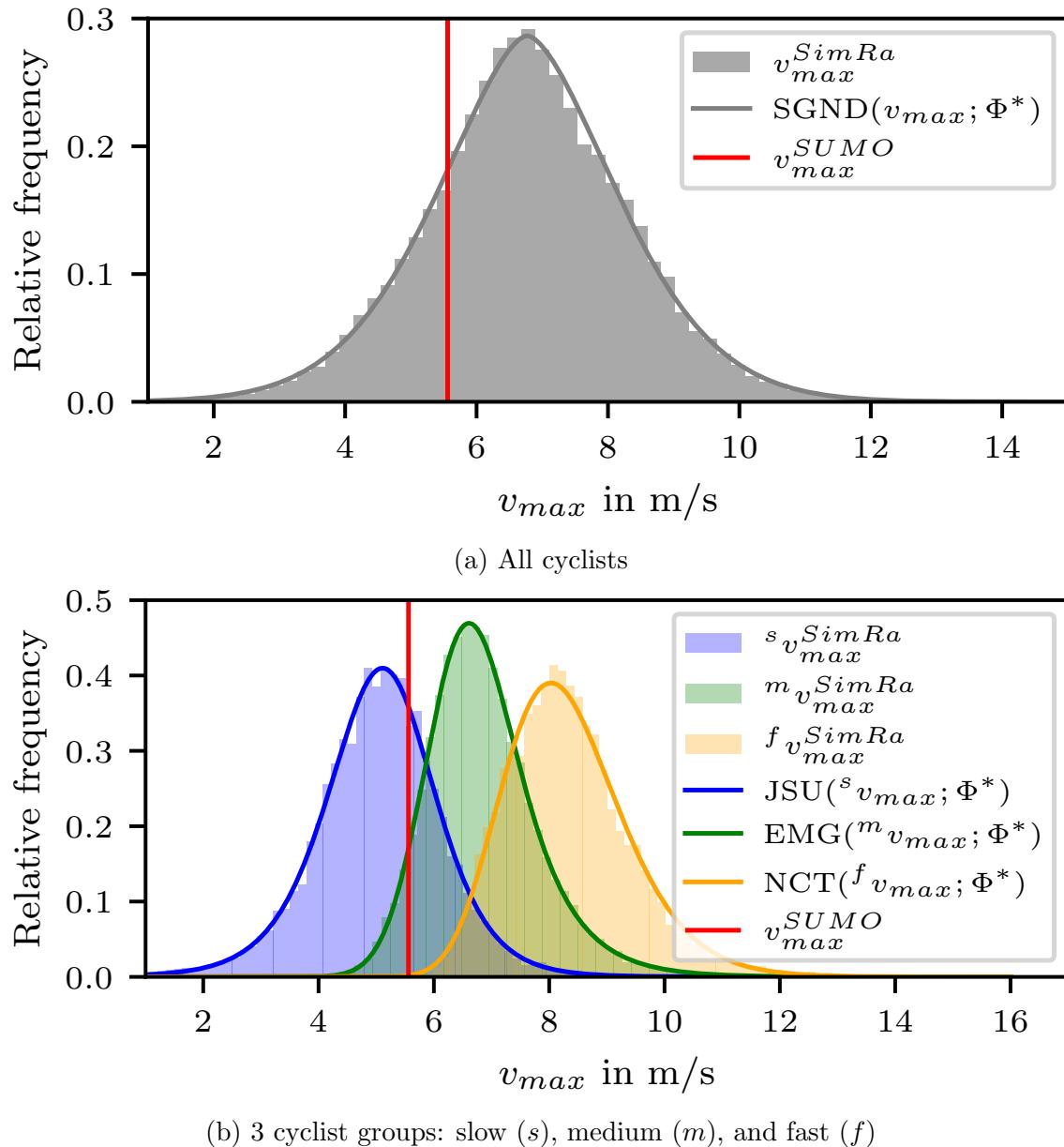


Figure 3.21: Histogram of maximum velocity capabilities found in the empirical SimRa dataset and their respective distribution functions. The red scalar represents the default value in SUMO.

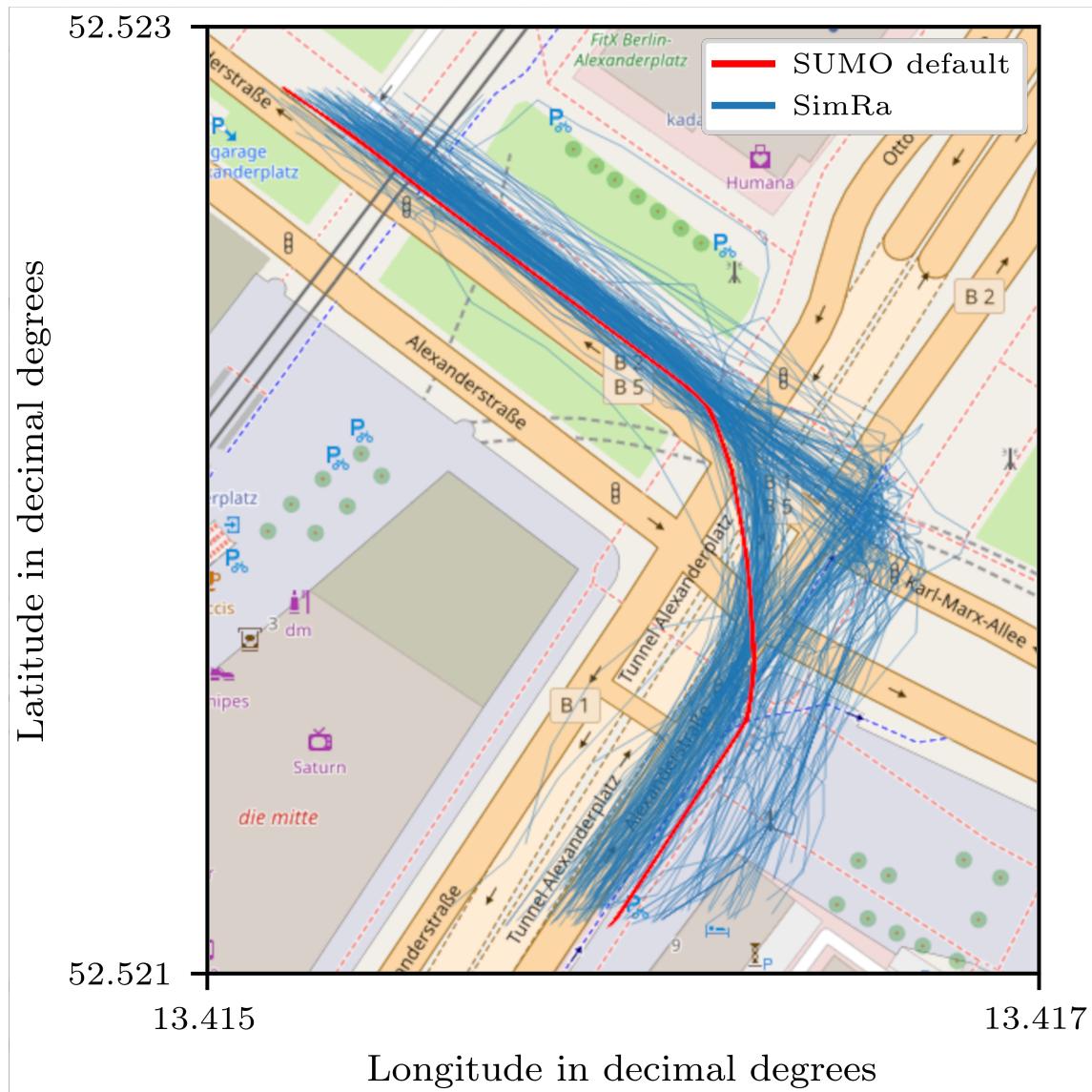


Figure 3.22: Qualitative comparison between the SUMO default intersection model and real world data given by SimRa for the intersection between Alexanderstraße and Karl-Marx-Allee in Berlin. SimRa shows two distinct left-turn paths (i.e., a direct and an indirect one) whereas SUMO default only models the direct path.

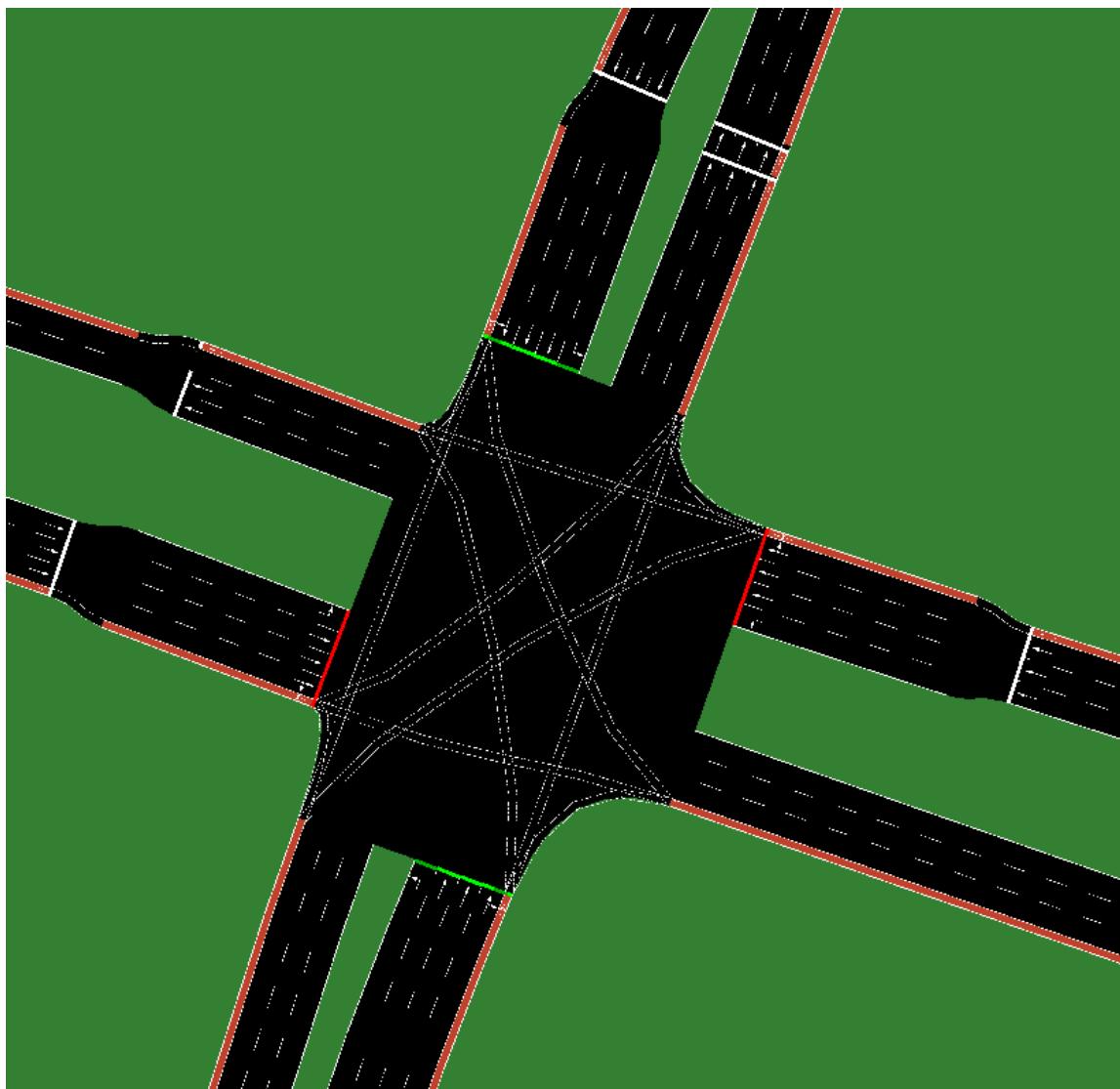
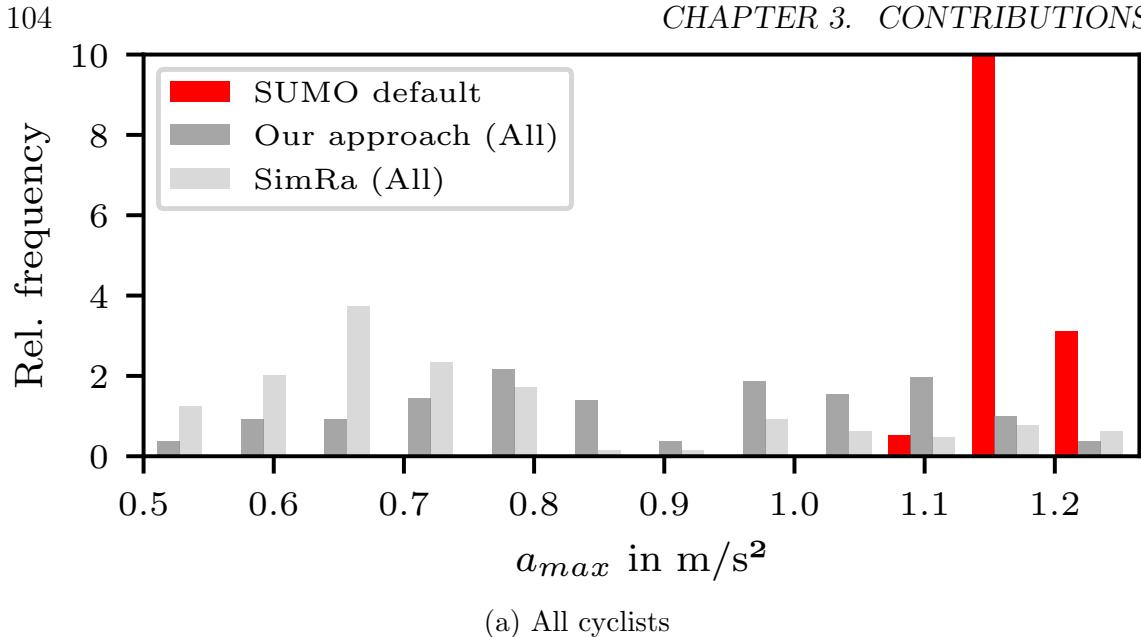
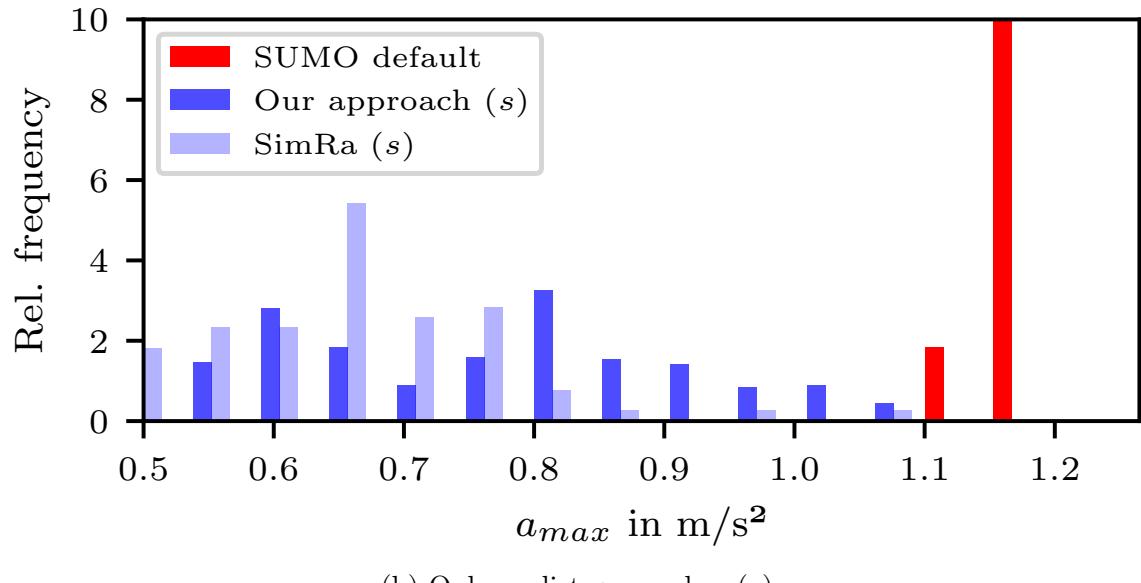


Figure 3.23: Excerpt from the *Mehringdamm* scenario in SUMO. The scenario was created using OSM data only.



(a) All cyclists



(b) Only cyclist group slow (s)

Figure 3.24: Histogram of SUMO's, SimRa's, and our approach's observed maximum accelerations inside the example *Frauentorgraben* scenario. While the maximum accelerations are heterogeneously distributed in the real-world data and our approach, the default values are clustered. Here, the observed accelerations deviate from the configured default value (1.2 m/s^2) due to traffic effects inside the simulation.

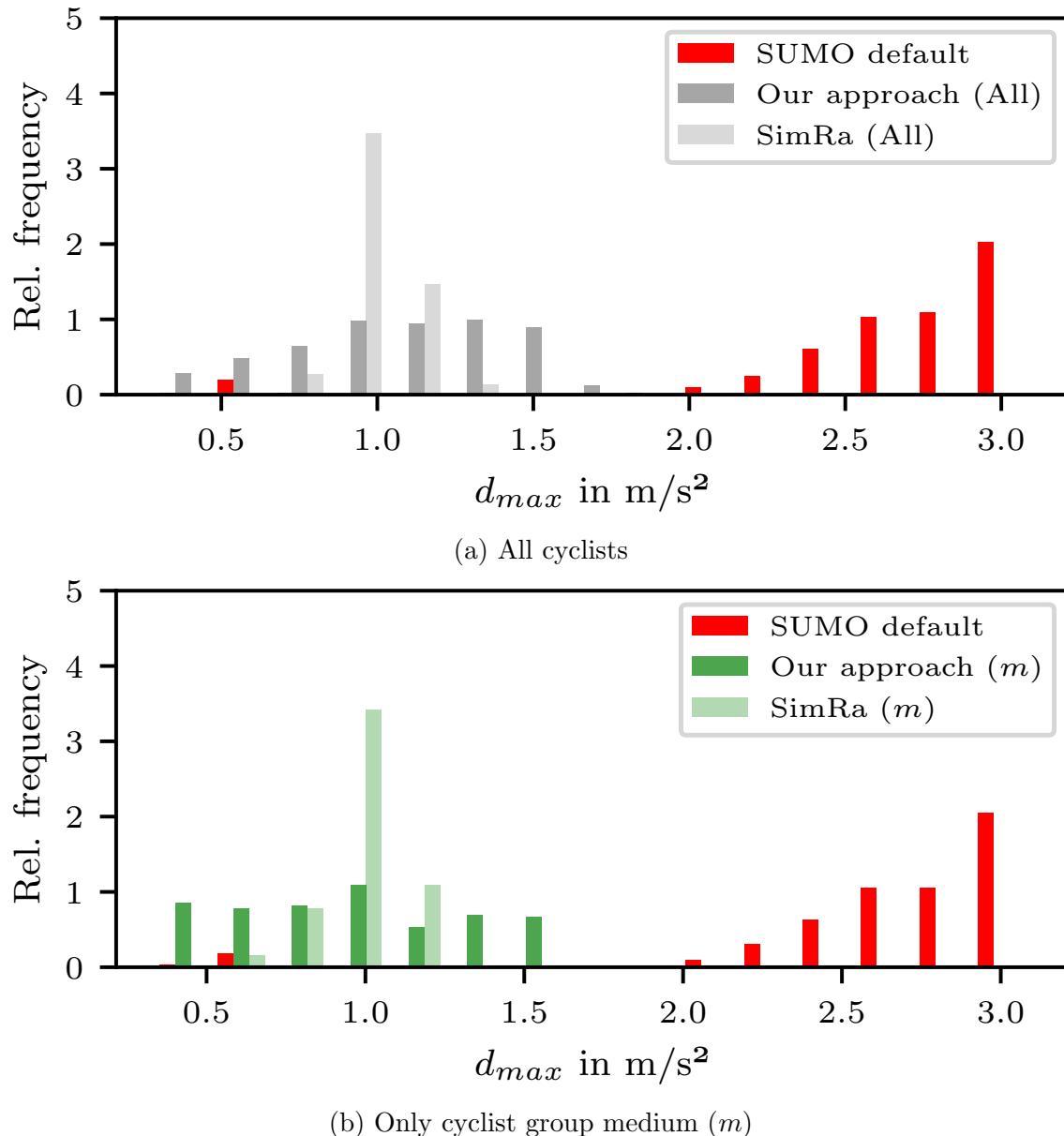


Figure 3.25: Histogram of SUMO's, SimRa's, and our approach's observed maximum decelerations inside the example *Dachauer Straße* scenario. While the maximum decelerations are heterogeneously distributed in the real-world data and our approach, the default values are clustered. Here, the observed decelerations deviate from the configured default value (3m/s^2) due to traffic effects inside the simulation.

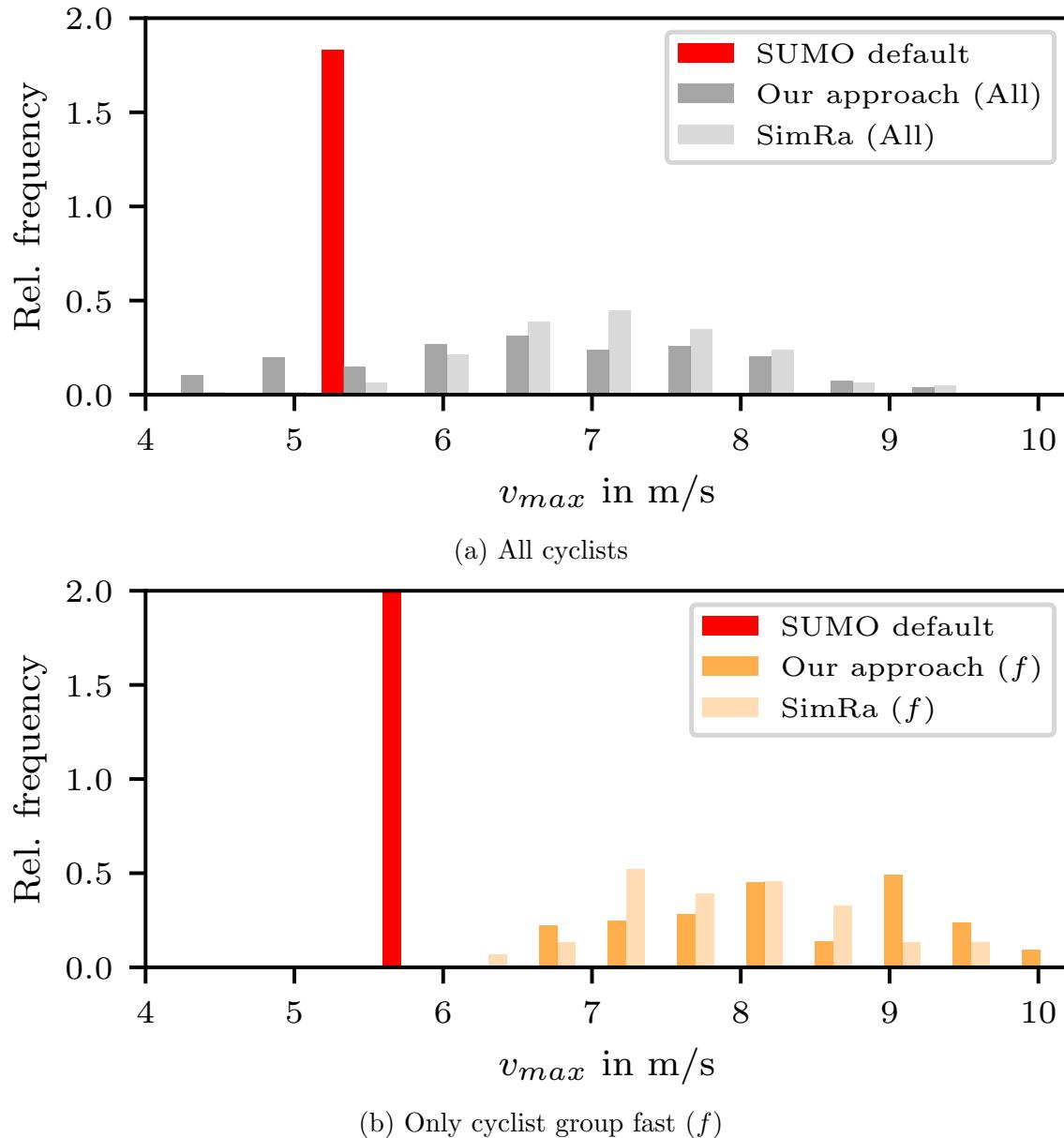


Figure 3.26: Histogram of SUMO’s, SimRa’s, and our approach’s observed maximum velocities inside the example *Oranienstraße* scenario. While the maximum velocities are heterogeneously distributed in the real-world data and our approach, the default values are clustered.

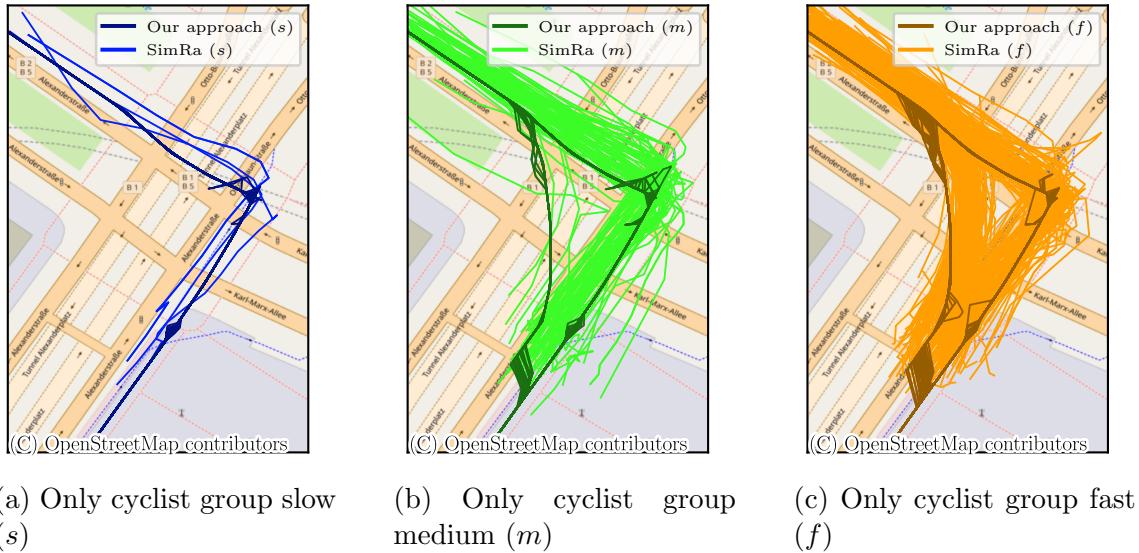


Figure 3.27: Qualitative comparison between the results of our approach for the intersection model and real-world raw GPS data given by SimRa for the intersection between Alexanderstraße and Karl-Marx-Allee in Berlin. SimRa shows two distinct left-turn paths (i.e., a direct and an indirect one), which are also modeled by our models. Also the three different groups have different shares between direct and indirect left-turns.

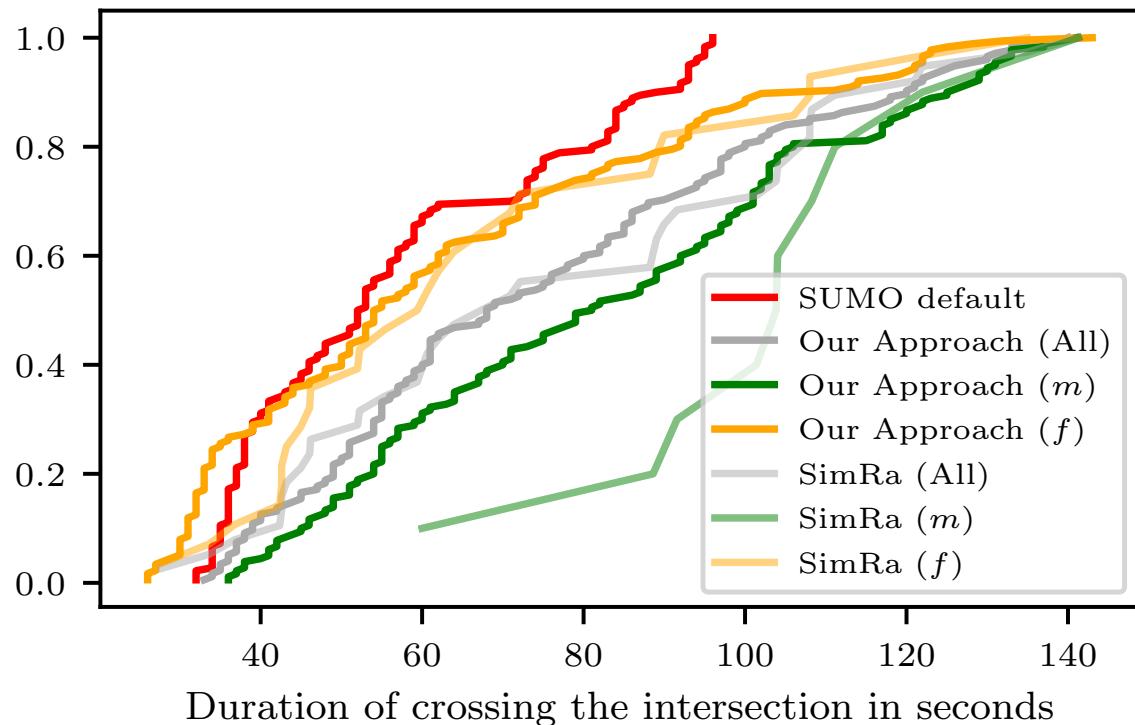


Figure 3.28: ECDFs of the measured durations for crossing the scenario *Warschauer Straße*. It is apparent that our models for all, medium (m) and fast (f) cyclists outperform SUMO's default as the measured durations converge towards the real-world data. Only our fast cyclist model (f) is just marginally better than SUMO's default.

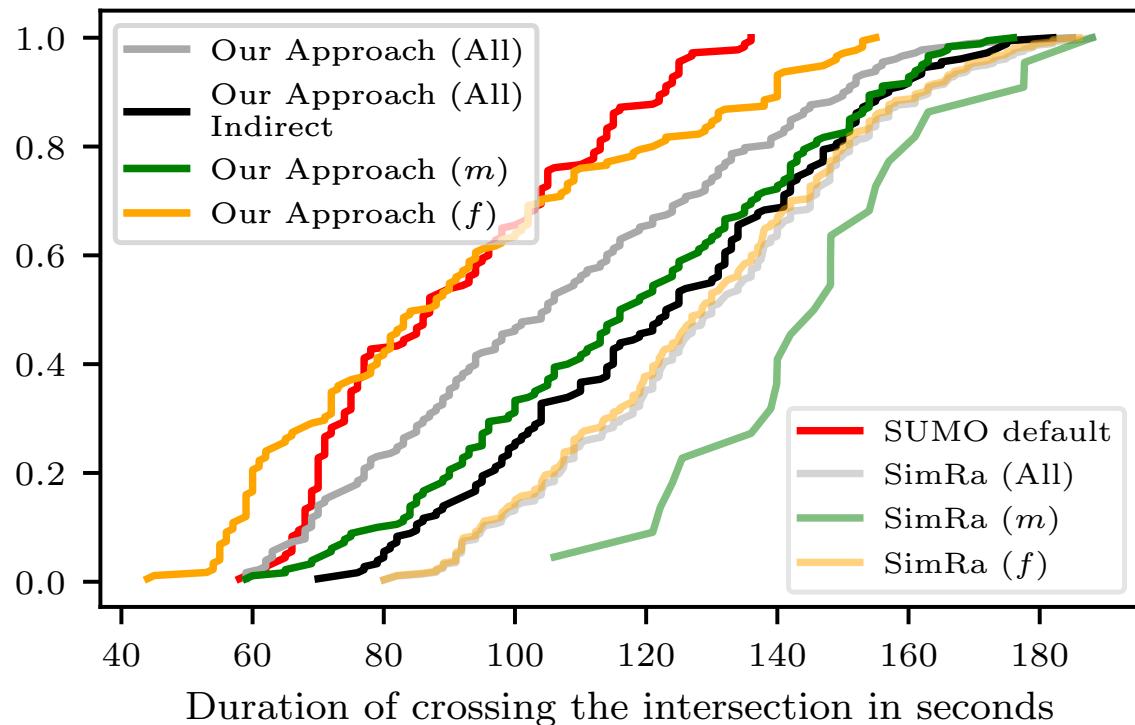


Figure 3.29: ECDFs of the measured durations for crossing the scenario *Mehringdamm*. The results when using our models for all, medium (m) and fast (f) cyclists are only slightly more realistic than when using the standard SUMO model. However, when the *direct* path is blocked for cyclists, the simulation results outperform the default approach.

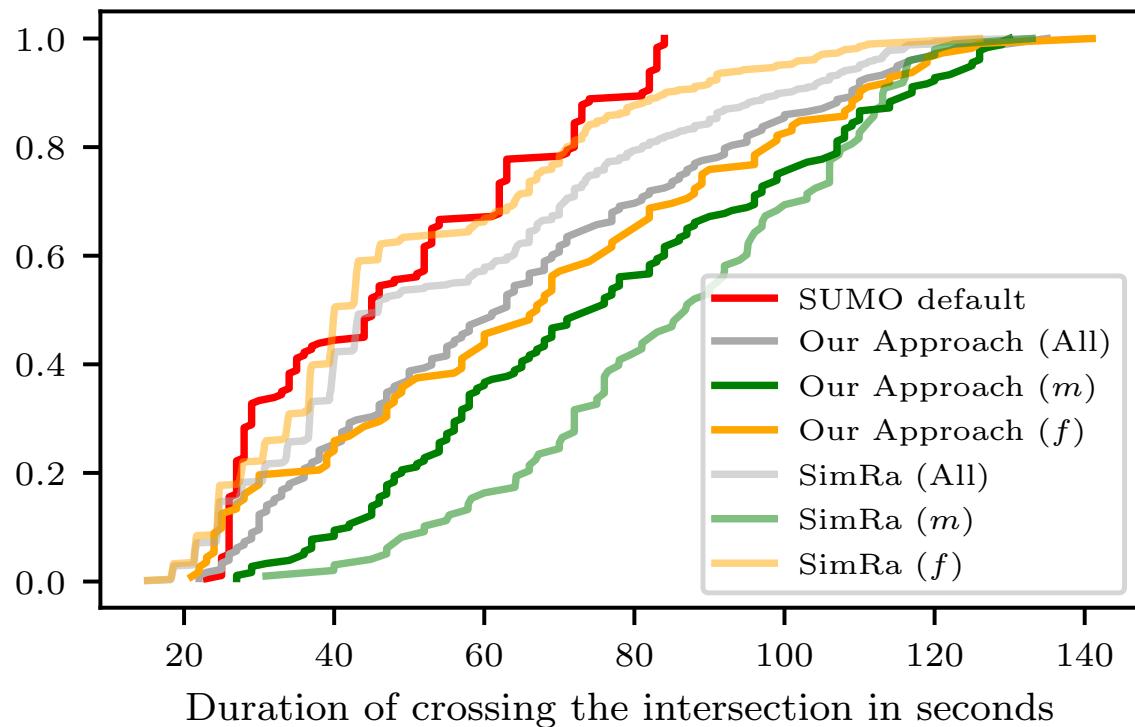


Figure 3.30: ECDFs of the measured durations for crossing the scenario *Alexanderstraße*. Here, our new approach is not more precise than SUMO default behavior.



Figure 3.31: Intersection Mehringdamm/Gneisenaustraße: a traffic island obstructs the direct turn path (dashed line) and, thus, makes the indirect path (solid line) more likely to be used.

Chapter 4

Conclusions

4.1 Discussion

4.2 Summary and Outlook

Appendix A

Technologies and Concepts of a Larger Context

This appendix chapter contains definitions for technologies and concepts that are mentioned in the thesis, but are not of higher importance for it. Section [A.1](#) introduces a consensus algorithm for distributed systems.

A.1 Finding Consensus in a Distributed System

The Paxos algorithm has first been described by Leslie Lamport in their work about the parliament on the island of Paxos.

Appendix B

Acronyms

AI Artificial Intelligence

ANN Artificial Neural Network

AUC Area under the curve

MCC Matthews correlation coefficient

BCE Binary Cross-Entropy

CNN Convolutional Neural Network

DBN Deep Belief Network

DFT Discrete Fourier Transform

DL Deep Learning

DNN Deep Neural Network

ECG Electrocardiogram

ESN Echo State Network

FCN Fully Connected Network

FPR False Positive Rate

GAF Gramian Angular Field

GAN Generative Adversarial Network

GDPR General Data Protection Regulation

GPS Global Positioning System

GRU Gated Recurrent Unit

HAR Human Activity Recognition

IQR Interquartile Range

LSTM Long Short-Term Memory

ML Machine Learning

MLP Multi-Layer Perceptron

NLP Natural Language Processing

OBS OpenBikeSensor

OSM OpenStreetMap

ROC Receiver Operating Characteristic

RNN Recurrent Neural Network

SAS Sensitivity at Specificity

SDAE Stacked Denoising Auto Encoder

SF Sensor-based Fusion

SGD Stochastic Gradient Descent

SVM Support Vector Machine

TPR True Positive Rate

TSC Time Series Classification

XAI Explainable Artificial Intelligence

Appendix C

Lexicon

Byzantine failure A malfunction of a component that leads to the distribution of wrong/conflicting information to other parts of the system is called Byzantine failure. The name is based on the Byzantines Generals Problem, in which three Byzantine generals need to agree on a battle plan while one or more of them might be a traitor trying to confuse the others.

Appendix D

Listings

This is the appendix for code, that does not need to be provided directly inside the thesis.

D.1 Configuration for Node A

Listing D.1: Configuration for Node A

```
{  
    "nodeID" : "nodeA",  
    "publicKey" : "<public key>",  
    "encryptionAlgorithm" : "RSA",
```

```
    "machines" : [ "192.168.0.132", "192.168.0.165" ] ,  
    "publisherPort" : 8000 ,  
    "messagePort" : 8010 ,  
    "restPort" : 8080 ,  
    "location" : "52.515249, 13.326476" ,  
    "description" : "Raspberry Pi Cluster in EN 004"  
}
```

Bibliography

- [1] Moloud Abdar et al. “A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges”. In: *Information Fusion* 76 (Dec. 2021), pp. 243–297.
- [2] Munshi Yusuf Alam et al. “Crowdsourcing from the True crowd: Device, vehicle, road-surface and driving independent road profiling from smartphone sensors”. In: *Pervasive and Mobile Computing* 61 (2020), p. 101103.
- [3] Rachel Aldred and Anna Goodman. “Predictors of the frequency and subjective experience of cycling near misses: Findings from the first two years of the UK Near Miss Project”. In: *Accident Analysis & Prevention* 110 (2018).
- [4] Davide Anguita et al. “A public domain dataset for human activity recognition using smartphones”. In: *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*. 2013.

- [5] M. Bando et al. “Dynamical model of traffic congestion and numerical simulation”. In: *APS Physical Review E (PRE)* 51.2 (Feb. 1995), pp. 1035–1042. ISSN: 2470-0053. DOI: [10.1103/physreve.51.1035](https://doi.org/10.1103/physreve.51.1035).
- [6] Anahid Basiri, Muki Haklay, and Zoe Gardner. “The Impact of Biases in the Crowdsourced Trajectories on the Output of Data Mining Processes”. In: *21th AGILE International Conference on Geographic Information Science (AGILE 2018), VGI-ALIVE - AnaLysis, Integration, Vision, Engagement*. Lund, Sweden, June 2018.
- [7] Anahid Basiri et al. *Crowdsourced geospatial data quality: Challenges and future directions*. 2019.
- [8] Stephen Bates et al. “Testing for Outliers with Conformal p-values”. In: *arXiv:2104.08279 [math, stat]* (Apr. 2021).
- [9] Jossekin Beilharz et al. “Towards a Staging Environment for the Internet of Things”. In: *19th IEEE International Conference on Pervasive Computing and Communications (PerCom 2021), PerCom Work in Progress on Pervasive Computing and Communications (PerCom Workshops)*. Virtual Conference: IEEE, Mar. 2021, pp. 312–315. ISBN: 978-1-66540-424-2. DOI: [10.1109/percomworkshops51409.2021.9431087](https://doi.org/10.1109/percomworkshops51409.2021.9431087).
- [10] David Bermbach. “SimRa Rides Berlin 06/19 - 12/20”. en. In: Technische Universität Berlin, 2021. DOI: [10.14279/depositonce-10605](https://doi.org/10.14279/depositonce-10605).

- [11] David Bermbach and Ahmet-Serdar Karakaya. “SimRa Rides Berlin 01/21 - 09/21”. en. In: Technische Universität Berlin, 2021. doi: [10.14279/depositonce-12452](https://doi.org/10.14279/depositonce-12452).
- [12] Felix Biessmann et al. “Automated data validation in machine learning systems”. In: *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* (2021).
- [13] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [14] Bryan Blanc and Miguel Figliozzi. “Modeling the impacts of facility type, trip characteristics, and trip stressors on cyclists’ comfort levels utilizing crowd-sourced data”. In: *Transportation Research Record* 2587.1 (2016).
- [15] Bryan Blanc and Miguel Figliozzi. *Safety perceptions, roadway characteristics, and cyclists’ demographics: A study of crowdsourced smartphone bicycle safety data*. Tech. rep. 2017.
- [16] Eric Breck et al. “Data Validation for Machine Learning”. In: *SysML*. 2019, pp. 1–14.
- [17] Andreas Bulling, Ulf Blanke, and Bernt Schiele. “A tutorial on human activity recognition using body-worn inertial sensors”. In: *ACM Computing Surveys* 46.3 (2014).

- [18] Irving W. Burr. “Cumulative Frequency Functions”. In: *The Annals of Mathematical Statistics* 13.2 (June 1942), pp. 215–232. ISSN: 0003-4851. DOI: [10.1214/aoms/1177731607](https://doi.org/10.1214/aoms/1177731607).
- [19] Abhijnan Chakraborty et al. “Who makes trends? understanding demographic biases in crowdsourced recommendations”. In: *AAAI ICWSM 2017*. Vol. 11. 1. 2017.
- [20] Robert E. Chandler, Robert Herman, and Elliott W. Montroll. “Traffic Dynamics: Studies in Car Following”. In: *INFORMS Operations Research* 6.2 (Apr. 1958), pp. 165–184. DOI: [10.1287/inro.6.2.165](https://doi.org/10.1287/inro.6.2.165).
- [21] Kaixuan Chen et al. “Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities”. In: *ACM Computing Surveys* 54.4 (2021).
- [22] Kongyang Chen et al. “CRSM: Crowdsourcing based road surface monitoring”. In: *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing*. IEEE. 2013, pp. 2151–2158.
- [23] Arijit Chowdhury, Tapas Chakravarty, and P Balamuralidhar. “Estimating true speed of moving vehicle using smartphone-based GPS measurement”. In: *IEEE International Conference on Systems, Man and Cybernetics (SMC 2014)*. San Diego, CA: IEEE, Oct. 2014. DOI: [10.1109/smcc.2014.6974444](https://doi.org/10.1109/smcc.2014.6974444).
- [24] Junyoung Chung et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling”. In: *arXiv preprint arXiv:1412.3555* (2014).

- [25] Ekin D. Cubuk et al. “Autoaugment: Learning augmentation strategies from data”. In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* Vol. 2019-June. May 2019, pp. 113–123.
- [26] Steven L Cumbaa. *Road Profile Study. Final Report*. Tech. rep. 1986.
- [27] Yousef-Awwad Daraghmi, Tsung-Hsiang Wu, and Tsì-Uí Ík. “Crowdsourcing-based road surface evaluation and indexing”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.5 (2020), pp. 4164–4175.
- [28] Andreas Eckner. “A framework for the analysis of unevenly spaced time series data”. In: *Preprint. Available at: http://www.eckner.com/papers/unevenly_spaced_time_series_analysis* (2012).
- [29] Wilfried Elmenreich. “Sensor fusion in time-triggered systems”. PhD thesis. Citeseer, 2002.
- [30] N. Etemadi. “An elementary proof of the strong law of large numbers”. In: *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 55.1 (Feb. 1981), pp. 119–122. ISSN: 1432-2064. DOI: [10.1007/bf01013465](https://doi.org/10.1007/bf01013465).
- [31] Tom Fawcett. “An introduction to ROC analysis”. In: *Pattern recognition letters* 27.8 (2006).
- [32] April Gadsby, James Tsai, and Kari Watkins. “Understanding the Influence of Pavement Conditions on Cyclists’ Perception of Safety and Comfort Using Surveys and Eye Tracking”. In: *Transportation Research Record* 2676.12 (2022), pp. 112–126. DOI: [10.1177/03611981221090936](https://doi.org/10.1177/03611981221090936).

- [33] Denos C. Gazis, Robert Herman, and Richard W. Rothery. “Nonlinear Follow-The-Leader Models of Traffic Flow”. In: *INFORMS Operations Research* 9.4 (Aug. 1961), pp. 545–567. DOI: [10.1287/opre.9.4.545](https://doi.org/10.1287/opre.9.4.545).
- [34] Peter G. Gipps. “A Behavioural Car-Following Model for Computer Simulation”. In: *Elsevier Transportation Research Part B: Methodological* 15.2 (Apr. 1981), pp. 105–111. ISSN: 0191-2615. DOI: [10.1016/0191-2615\(81\)90037-0](https://doi.org/10.1016/0191-2615(81)90037-0).
- [35] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [36] Thomas Götschi, Jan Garrard, and Billie Giles-Corti. “Cycling as a part of daily life: a review of health perspectives”. In: *Taylor & Francis Transport Reviews* 36.1 (2016), pp. 45–71. DOI: [10.1080/01441647.2015.1057877](https://doi.org/10.1080/01441647.2015.1057877).
- [37] Georgios Grigoropoulos et al. “Modelling Bicycle Infrastructure in SUMO”. In: *SUMO User Conference 2019 (SUMO 2019)*. Ed. by Melanie Weber et al. Berlin, Germany: EasyChair, Aug. 2019, pp. 187–198. DOI: [10.29007/6cs5](https://doi.org/10.29007/6cs5).
- [38] Georgios Grigoropoulos et al. “Traffic flow at signalized intersections with large volumes of bicycle traffic”. In: *Elsevier Transportation Research Part A: Policy and Practice* 155 (Jan. 2022), pp. 464–483. ISSN: 0965-8564. DOI: [10.1016/j.tra.2021.11.021](https://doi.org/10.1016/j.tra.2021.11.021).
- [39] Eli. Grushka. “Characterization of exponentially modified Gaussian peaks in chromatography”. In: *ACS Analytical Chemistry* 44.11 (Sept. 1972), pp. 1733–1738. DOI: [10.1021/ac60319a011](https://doi.org/10.1021/ac60319a011).

- [40] Takaki Hayashi and Nakahiro Yoshida. “On covariance estimation of non-synchronously observed diffusion processes”. In: *Bernoulli* 11.2 (2005).
- [41] Kaiming He et al. “Deep residual learning for image recognition”. In: *IEEE CVPR 2016*. 2016.
- [42] Julian Heinovski et al. “Modeling Cycling Behavior to Improve Bicyclists’ Safety at Intersections – A Networking Perspective”. In: *20th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2019)*. Washington, D.C.: IEEE, June 2019. ISBN: 978-1-7281-0270-2. DOI: [10.1109/WoWMoM.2019.8793008](https://doi.org/10.1109/WoWMoM.2019.8793008).
- [43] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997).
- [44] D. Hogben, R. S. Pinkham, and M. B. Wilk. “The moments of the non-central t-distribution”. In: *OUP Biometrika* 48.3-4 (1961), pp. 465–468. DOI: [10.1093/biomet/48.3-4.465](https://doi.org/10.1093/biomet/48.3-4.465).
- [45] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. PMLR. 2015.
- [46] Hassan Ismail Fawaz et al. “Deep learning for time series classification: a review”. In: *Data mining and knowledge discovery* 33.4 (2019).

- [47] Norman L. Johnson. “Systems of Frequency Curves Generated by Methods of Translation”. In: *MAA Biometrika* 36.1/2 (June 1949), p. 149. ISSN: 1464-3510. DOI: [10.2307/2332539](https://doi.org/10.2307/2332539).
- [48] Daniel Kahneman, Jack L Knetsch, and Richard H Thaler. “Anomalies: The endowment effect, loss aversion, and status quo bias”. In: *Journal of Economic perspectives* 5.1 (1991).
- [49] Ahmet-Serdar Karakaya and David Bermbach. “SimRa Rides Berlin 10/21 - 09/22”. en. In: Technische Universität Berlin, 2022. DOI: [10.14279/depositonce-16439](https://doi.org/10.14279/depositonce-16439).
- [50] Ahmet-Serdar Karakaya, Jonathan Hasenburg, and David Bermbach. “SimRa: Using crowdsourcing to identify near miss hotspots in bicycle traffic”. In: *PMC* 67 (2020).
- [51] Ahmet-Serdar Karakaya et al. “A Realistic Cyclist Model for SUMO Based on the SimRa Dataset”. In: *20th IEEE Mediterranean Communication and Computer Networking Conference (MedComNet 2022)*. Paphos, Cyprus: IEEE, June 2022, pp. 166–173. DOI: [10.1109/MedComNet55087.2022.9810439](https://doi.org/10.1109/MedComNet55087.2022.9810439).
- [52] Ahmet-Serdar Karakaya et al. “CycleSense: Detecting Near Miss Incidents in Bicycle Traffic from Mobile Motion Sensors”. In: *Elsevier Pervasive and Mobile Computing* (2023).
- [53] Heather Kath and Georgios Grigoropoulos. “Integration of an External Bicycle Model in SUMO”. In: *SUMO User Conference 2016 (SUMO 2016)*. Vol. 30. Berlin, Germany: DLR, May 2016.

- [54] Stefan Krauß. “Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics”. PhD Thesis. Köln, Germany: Mathematical Institute, Apr. 1998.
- [55] Tim Kuhlmann, Pablo Garaizar, and Ulf-Dietrich Reips. “Smartphone sensor accuracy varies from device to device in mobile research: The case of spatial orientation”. In: *Behavior research methods* 53.1 (2021).
- [56] W Leutzbach and R Wiedemann. “Development and applications of traffic simulation models at the Karlsruhe Institut für Verkehrswesen”. In: *Traffic engineering & control (TEC)* 27.5 (1986), pp. 270–278. ISSN: 0041-0683.
- [57] Xiao Li, Ruizhi Chen, and Tianxing Chu. “A crowdsourcing solution for road surface roughness detection using smartphones”. In: *Proceedings of the 27th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2014)*. 2014, pp. 498–502.
- [58] Lucas Cedro Lima et al. “Using crowdsourcing techniques and mobile devices for asphaltic pavement quality recognition”. In: *2016 VI Brazilian symposium on computing systems engineering (SBESC)*. IEEE. 2016, pp. 144–149.
- [59] Zachary C. Lipton, Yu Xiang Wang, and Alexander J. Smola. “Detecting and correcting for label shift with black box predictors”. In: *35th Int. Conf. Mach. Learn. ICML 2018* 7 (Feb. 2018), pp. 4887–4897.
- [60] Pablo Alvarez Lopez et al. “Microscopic Traffic Simulation using SUMO”. In: *21st IEEE International Conference on Intelligent Transportation Systems*

- (ITSC 2018). Maui, HI: IEEE, Nov. 2018, pp. 2575–2582. ISBN: 978-1-7281-0323-5. DOI: [10.1109/itsc.2018.8569938](https://doi.org/10.1109/itsc.2018.8569938).
- [61] Kai Luedemann and Mario A. Nascimento. “BikeVibes: An App for Crowd-sourcing Open Road Quality Data from a Cyclist Perspective”. In: *Proceedings of the 15th ACM SIGSPATIAL International Workshop on Computational Transportation Science*. IWCTS ’22. Seattle, Washington: Association for Computing Machinery, 2022. ISBN: 9781450395397. DOI: [10.1145/3557991.3567779](https://doi.org/10.1145/3557991.3567779). URL: <https://doi.org/10.1145/3557991.3567779>.
- [62] Xiaoliang Ma and Ding Luo. “Modeling cyclist acceleration process for bicycle traffic simulation using naturalistic data”. In: *Elsevier Transportation Research Part F: Traffic Psychology and Behaviour* 40 (July 2016), pp. 130–144. ISSN: 1369-8478. DOI: [10.1016/j.trf.2016.04.009](https://doi.org/10.1016/j.trf.2016.04.009).
- [63] Krista Merry and Pete Bettinger. “Smartphone GPS accuracy study in an urban environment”. In: *PLOS ONE* 14.7 (July 2019), pp. 1–19. DOI: [10.1371/journal.pone.0219890](https://doi.org/10.1371/journal.pone.0219890).
- [64] Paul W. Mielke. “Another Family of Distributions for Describing and Analyzing Precipitation Data”. In: *AMS Journal of Applied Meteorology and Climatology (JAMC)* 12.2 (1973), pp. 275–280. ISSN: 1558-8432. DOI: [10.1175/1520-0450\(1973\)012<0275:AFODFD>2.0.CO;2](https://doi.org/10.1175/1520-0450(1973)012<0275:AFODFD>2.0.CO;2).
- [65] Saralees Nadarajah. “A generalized normal distribution”. In: *Taylor & Francis Journal of Applied Statistics (JAS)* 32.7 (Sept. 2005), pp. 685–694. DOI: [10.1080/02664760500079464](https://doi.org/10.1080/02664760500079464).

- [66] Peter Nyberg, Ulf Björnstag, and L-O Bygren. “Road characteristics and bicycle accidents”. In: *Scandinavian Journal of Social Medicine* 24.4 (1996), pp. 293–301.
- [67] Marie-Christin H. Oczko et al. “Integrating Haptic Signals with V2X-based Safety Systems for Vulnerable Road Users”. In: *IEEE International Conference on Computing, Networking and Communications (ICNC 2020)*. Kailua, HI: IEEE, Feb. 2020, pp. 692–697. ISBN: 978-1-7281-4905-9. DOI: [10.1109/ICNC47757.2020.9049723](https://doi.org/10.1109/ICNC47757.2020.9049723).
- [68] Javier Ortiz Laguna, Angel Garcia Olaya, and Daniel Borrajo. “A dynamic sliding window approach for activity recognition”. In: *UMAP 2011*. Springer. 2011.
- [69] Felipe E. Pedroso et al. “Bicycle Use and Cyclist Safety Following Boston’s Bicycle Infrastructure Expansion, 2009–2012”. In: *American Journal of Public Health (AJPH)* 106.12 (Dec. 2016), pp. 2171–2177. DOI: [10.2105/ajph.2016.303454](https://doi.org/10.2105/ajph.2016.303454).
- [70] Daiyan Peng et al. “A Road Condition Classifier via Lock Embedded IMU on Dock-Less Shared Bikes”. In: *Proceedings of the International Conference on Industrial Control Network and System Engineering Research*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 32–36. ISBN: 9781450366274.
- [71] Neoklis Polyzotis et al. “Data lifecycle challenges in production machine learning: A survey”. In: *SIGMOD Rec.* Vol. 47. 2018, pp. 17–28.

- [72] Stephan Rabanser, Stephan Günemann, and Zachary C. Lipton. *Failing loudly: An empirical study of methods for detecting dataset shift*. 2018.
- [73] Charissa Ann Ronao and Sung-Bae Cho. “Deep Convolutional Neural Networks for Human Activity Recognition with Smartphone Sensors”. In: *Neural Information Processing*. Springer International Publishing, 2015.
- [74] Dominik Salles, Stefan Kaufmann, and Hans-Christian Reuss. “Extending the Intelligent Driver Model in SUMO and Verifying the Drive Off Trajectories with Aerial Measurements”. In: *SUMO User Conference 2020 (SUMO 2020)*. Virtual Conference, Oct. 2020. doi: [10.5281/ZENODO.5175999](https://doi.org/10.5281/ZENODO.5175999).
- [75] Albert Sánchez Fuster. “Detecting Near Miss Incidents in Bicycle Traffic Using Acceleration Sensor Data”. MA thesis. Universitat Politècnica de Catalunya, 2020.
- [76] Michael W. Sayers. “The International Road Roughness Experiment (IRRE) : establishing correlation and a calibration standard for measurements”. In: 1986.
- [77] Sebastian Schelter, Tammo Rukat, and Felix Biessmann. “Learning to Validate the Predictions of Black Box Classifiers on Unseen Data”. In: *Proc. 2020 ACM SIGMOD Int. Conf. Manag. Data*. New York, NY, USA: ACM, June 2020, pp. 1289–1299.
- [78] Sebastian Schelter et al. “On Challenges in Machine Learning Model Management”. In: *Bull. IEEE Comput. Soc. Tech. Comm. Data Eng.* (2018), pp. 5–13.

- [79] Allan Stisen et al. “Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition”. In: *ACM SenSys 2015*. 2015.
- [80] Student. “The Probable Error of a Mean”. In: *MAA Biometrika* 6.1 (Mar. 1908), pp. 1–25. ISSN: 1464-3510. DOI: [10.2307/2331554](https://doi.org/10.2307/2331554).
- [81] M Sugiyama and Motoaki Kawanabe Motoaki. *Machine Learning in Non-Stationary Environments Introduction to Covariate Shift Adaptation*. MIT Press, 2012.
- [82] Yoshiaki Taniguchi, Kodai Nishii, and Hiroyuki Hisamatsu. “Evaluation of a Bicycle-Mounted Ultrasonic Distance Sensor for Monitoring Road Surface Condition”. In: *2015 7th International Conference on Computational Intelligence, Communication Systems and Networks*. 2015, pp. 31–34.
- [83] Dapeng Tao, Yonggang Wen, and Richang Hong. “Multicolumn Bidirectional Long Short-Term Memory for Mobile Devices-Based Human Activity Recognition”. In: *IEEE Internet of Things Journal* 3.6 (2016).
- [84] Waldemar Titov and Thomas Schlegel. “Monitoring Road Surface Conditions for Bicycles – Using Mobile Device Sensor Data from Crowd Sourcing”. In: June 2019, pp. 340–356. ISBN: 978-3-030-22665-7. DOI: [10.1007/978-3-030-22666-4_25](https://doi.org/10.1007/978-3-030-22666-4_25).
- [85] Du Tran et al. “Learning spatiotemporal features with 3d convolutional networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2015.

- [86] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. “Congested Traffic States in Empirical Observations and Microscopic Simulations”. In: *APS Physical Review E (PRE)* 62.2 (Aug. 2000), pp. 1805–1824. ISSN: 2470-0053. doi: [10.1103/PhysRevE.62.1805](https://doi.org/10.1103/PhysRevE.62.1805).
- [87] John W Tukey et al. *Exploratory data analysis*. Vol. 2. Reading, Mass., 1977.
- [88] Heather Twaddle and Georgios Grigoropoulos. “Modeling the Speed, Acceleration, and Deceleration of Bicyclists for Microscopic Traffic Simulation”. In: *Transportation Research Record (TRR)* 2587.1 (Jan. 2016), pp. 8–16. ISSN: 0361-1981. doi: [10.3141/2587-02](https://doi.org/10.3141/2587-02).
- [89] Graham Upton and Ian Cook. *Understanding statistics*. Oxford University Press, 1996.
- [90] Yuri Usami et al. “Bicycle Behavior Recognition using Sensors Equipped with Smartphone”. In: *8th IEEE International Conference on Consumer Electronics (ICCE 2018)*. Berlin, Germany: IEEE, Sept. 2018. doi: [10.1109/ICCE-Berlin.2018.8576254](https://doi.org/10.1109/ICCE-Berlin.2018.8576254).
- [91] Jindong Wang et al. “Deep learning for sensor-based activity recognition: A survey”. In: *Pattern Recognition Letters* 119 (2019).
- [92] Philip B Weerakody et al. “A review of irregular time series data handling with gated recurrent neural networks”. In: *Neurocomputing* 441 (2021).
- [93] David H Wolpert. “Stacked generalization”. In: *Neural networks* 5.2 (1992).

- [94] Jiahui Wu, Lingzi Hong, and Vanessa Frias-Martinez. “Predicting perceived cycling safety levels using open and crowdsourced data”. In: *IEEE Big Data 2018*. IEEE. 2018.
- [95] Takahiro Yamaguchi, Tomonori Nagayama, and Di Su. “Simple Estimation of Bicycle Lane Condition by Using the Dynamic Response of a Bicycle”. In: *6th International Conference on Advances in Experimental Structural Engineering and 11th International Workshop on Advanced Smart Materials and Smart Structures Technology (6AESE/11ANCRiSST)*. Aug. 2015.
- [96] Jianbo Yang et al. “Deep convolutional neural networks on multichannel time series for human activity recognition”. In: *IJCAI 2015*. 2015.
- [97] Shuochao Yao et al. “Deepsense: A unified deep learning framework for time-series mobile sensing data processing”. In: *TheWebConf 2017*. 2017.
- [98] William J Youden. “Index for rating diagnostic tests”. In: *Cancer* 3.1 (1950).
- [99] Baoding Zhou et al. “Smartphone-based road manhole cover detection and classification”. In: *Automation in Construction* 140 (2022), p. 104344.
- [100] Daniel Zwillinger and Stephen Kokoska. *CRC standard probability and statistics tables and formulae*. Crc Press, 1999.