

Revealing Challenges and Opportunities in Cycling Safety Through Crowdsensing

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

Tag der Wissenschaftlicher Aussprache: 11.08.2024

April 2024

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

Ahmet-Serdar Karakaya

Abstract

An increased modal share of bicycle traffic is a key mechanism to reduce emissions and solve traffic-related problems. However, a lack of comfort and (perceived) safety keeps people from using their bikes more frequently. To improve comfort and safety in bicycle traffic, city planners need (a), an overview of accidents, near miss incidents, and bike routes, (b), an overview of the cycling infrastructure's maintenance state and (c), a way to test the planned changes in the bicycle infrastructure before implementing them in the real life. This needs, however, are currently not fulfilled satisfactory. In this thesis, we present three main contributions to solve these problems. We first show SimRa, a crowdsourcing-based citizen science project to gather records of cycling trips and near miss incidents in bicycle traffic. As part of this contribution, we also present CycleSense, an approach based on Deep Learning to automatically detect near miss incidents from recorded cycling trip data. The hazardous hotspots in bicycle traffic revealed with SimRa is beneficial for city planners, because if they know of a dangerous place for cyclists, they can implement changes to increase the safety there. Cyclists, on the other hand, can use this information to circumvent these spots. In our second contribution, we describe an approach to derive the road surface quality from cycling trip data. Our approach uses data produced by the inertial measurement unit of the smartphone to calculate the smoothness of the road surface by the vibrations caused by traversing on the road with a bicycle. We then show how our approach can easily be integrated into a cycling trip recording app with SimRa as a case study. This contribution also contains a visualization of the cycling infrastructure's smoothness and helps both city planners and cyclists the same way as our first contribution. As our third contribution, we improve the cyclist simulation model of the urban traffic simulation software called SUMO. For this, we first show that the default cyclist model of SUMO is not realistic by comparing it to how cyclists behave through the SimRa dataset. We then split the cycling trips in the SimRa dataset into three parts for slow, medium and fast cyclists and implement our findings as a plugin that can be added to SUMO for a more realistic cyclist simulation. This can provide a strong tool for city planners to better test their planned changes in the traffic infrastructure.

Kurzdarstellung

Eine Erhöhung des Fahrradverkehrs trägt zur Verringerung der Treibhausgasemissionen und zur Lösung verkehrsbedingter Probleme bei. Mangelnder Komfort und (gefühlte) Unsicherheit halten die Menschen jedoch davon ab, öfter Fahrrad zu fahren. Um den Komfort und die Sicherheit im Radverkehr zu erhöhen, benötigen Stadtplaner (a) einen Überblick über Unfälle und Gefahrensituationen, (b) einen Überblick über den Wartungszustand der Radverkehrsinfrastruktur und (c) eine Möglichkeit, die geplanten Änderungen an der Radverkehrsinfrastruktur zu testen, bevor sie umgesetzt werden. In dieser Arbeit präsentieren wir drei Beiträge zur Lösung dieser Probleme. Zunächst stellen wir SimRa vor, ein Crowdsourcing-basiertes Citizen-Science-Projekt zur Erfassung Gefahrensituationen im Radverkehr. Im Rahmen dieses Beitrags stellen wir auch CycleSense vor, einen auf Deep Learning basierenden Ansatz zur automatischen Erkennung von Gefahrensituationen aus aufgezeichneten Radfaherdaten. Stadtplaner können die mit SimRa aufgedeckten gefährlichen Stellen im Radverkehr mit Baumaßnahmen verbessern. Radfahrer wiederum können diese Informationen nutzen, um diese Stellen zu umfahren. In unserem zweiten Beitrag beschreiben wir einen Ansatz zur Ableitung der Straßenoberflächenqualität aus Fahrradfahrtdataen. Unser Ansatz verwendet Daten, die von den Bewegungssensoren des Smartphones erzeugt werden, um Unebenheiten der Straßenoberfläche anhand der Vibrationen zu berechnen, die durch das Befahren der Straße mit einem Fahrrad verursacht werden. Wir zeigen dann, wie unser Ansatz in eine App zur Aufzeichnung von Fahrradtouren integriert werden kann, mit SimRa als Fallbeispiel. Dieser Beitrag enthält auch eine Visualisierung der Ergebnisse und hilft sowohl Stadtplanern als auch Radfahrern auf ähnliche Weise wie unser erster Beitrag. In unserem dritten Beitrag verbessern wir das Radfahrersimulationsmodell der Verkehrssimulationssoftware SUMO. Dazu zeigen wir zunächst, dass das Standard-Radfahrermodell von SUMO nicht realistisch ist, indem wir es mit dem Verhalten von Radfahrern anhand des SimRa-Datensatzes vergleichen. Anschließend teilen wir die Radfahrten im SimRa-Datensatz in drei Teile für langsame, mittlere und schnelle Radfahrer auf und implementieren unsere Ergebnisse als Plugin, das zu SUMO hinzugefügt werden kann, um eine realistischere Radfahrersimulation zu ermöglichen. Dies kann ein starkes Werkzeug für Stadtplaner sein, um ihre geplanten Änderungen an der Verkehrsinfrastruktur besser zu testen.

Table of Contents

Declaration	I
Abstract	II
Zusammenfassung	III
I Foundations	2
1 Introduction	3
1.1 Problem Statement	5
1.2 Contributions of this Thesis	6
1.2.1 SimRa: A Platform for Gathering Cycling Trip Data and Detecting Near Miss Incidents in Bicycle Traffic	6
1.2.2 Deriving Road Surface Quality from Cycling Trip Data	7
1.2.3 Cyclist Model for SUMO	7
1.3 Structure of this Thesis	8
2 Background and Related Work	9
2.1 Cycling Comfort	10

2.2	Cycling Safety	12
2.3	Crowdsourcing and Crowdsensing	14
2.4	Citizen Science	18
 II Improving Safety in Bicycle Traffic		 21
3	Gathering Cycling Trip Data and Detecting Incidents	23
3.1	Deep Learning for Time Series Classification	25
3.1.1	Time Series Classification	25
3.1.2	Deep Learning for Time Series Classification	25
3.2	SimRa - A Platform for Gathering Cycling Trips and Incidents in Bicycle Traffic . .	28
3.2.1	Goal of the SimRa platform	28
3.2.2	Data Acquisition	28
3.3	Detecting Incidents with Deep Learning	35
3.3.1	Preprocessing	35
3.3.2	Model Architecture	37
3.3.3	Model Training	39
3.4	Evaluation	41
3.4.1	Baselines	41
3.4.2	Metrics	42
3.4.3	Evaluation Results	42

<i>TABLE OF CONTENTS</i>	VI
3.5 Discussion	44
3.5.1 Impact of Preprocessing & Training Steps	45
3.5.2 Technical Limitations of the SimRa Data Set	45
3.5.3 Limitation of the Classification Task	46
3.5.4 Data Set Shift	48
3.6 Alternative Approaches	49
3.7 Summary	49
4 Deriving Road Surface Quality from Bicycle Ride Data	51
4.1 Road Surface Quality	52
4.2 Scenarios and Constraints	53
4.3 Integrating Road Surface Quality Measurement into SimRa	55
4.3.1 Overview of Pipeline	56
4.3.2 Surface Quality Analysis	56
4.4 Using Road Surface Quality Information	57
4.4.1 Routing with Surface Quality	57
4.4.2 Output Data and Visualization	57
4.5 Evaluation	58
4.5.1 Sections with Clear Results	59
4.5.2 Sections with Mixed Results	61
4.5.3 Sections with (Seemingly) Confusing Results	63

4.6	Discussion	65
4.6.1	Behavior of Cyclists	65
4.6.2	Sensor Inaccuracy	65
4.6.3	Temporal Influences	65
4.7	Alternative Approaches	66
4.8	Summary	67
5	Creating a Realistic Cyclist Model out of Bicycle Ride Data	68
5.1	Simulation of Urban Mobility (SUMO) - Simulation of Urban Mobility	69
5.2	Cycling Behavior in SimRa and SUMO	71
5.2.1	Preprocessing	72
5.2.2	Categorizing Cyclists by Velocity	72
5.2.3	Acceleration	74
5.2.4	Deceleration	75
5.2.5	Velocity	76
5.2.6	Left-turn Behavior at Intersections	77
5.3	Improving SUMO's Bicycle Simulation	78
5.3.1	Longitudinal Kinematic Behavior	78
5.3.2	Left-turn Behavior at Intersections	79
5.3.3	Different Cyclist Models	80
5.4	Evaluation	81

<i>TABLE OF CONTENTS</i>	VIII
5.4.1 Simulation Setup	81
5.4.2 Acceleration	83
5.4.3 Deceleration	85
5.4.4 Velocity	86
5.4.5 Left-turn Behavior at Intersections	88
5.4.6 Combining Intersection Model and Kinematic Extension	88
5.5 Discussion	90
5.5.1 Methodological Challenges	91
5.5.2 Dataset Choice as Ground Truth	92
5.5.3 Generalizability	93
5.5.4 E-bikes	93
5.6 Alternative Approaches	94
5.6.1 Intersection Behavior of Cyclists	94
5.6.2 Longitudinal Behavior of Cyclists	95
5.7 Summary	96
III Conclusions	97
6 Summary	99
7 Discussion and Outlook	103
7.1 Limitations of Crowdsourced Data	103

7.2 Limitations of the Preprocessing in SimRa	105
7.3 Outlook	106
A Technologies and Concepts of a Larger Context	108
A.1 Finding Consensus in a Distributed System	108
B Acronyms	109
C Lexicon	112
D Listings	113
D.1 Configuration for Node A	113
Bibliography	114

Part I

Foundations

Chapter 1

Introduction

The global climate crisis is and will be one of mankind's most vital challenges in the first half of the 21st century and the effects thereof are already observable. The frequency of extreme weather conditions and natural disasters such as long periods of precipitation leading to floods, or a lack of precipitation leading to droughts as well as heavy winds leading to storms is rising. Such disasters have short and long term effects on the agricultural production of the affected areas leading to famine, conflicts and mass migration. To prevent further aggravation of the effects, and stop or at least slow down the global warming, it is important to understand the root cause, so possible solutions can be derived and implemented. According to the Intergovernmental Panel on Climate Change ([IPCC](#)), the surface temperature of the globe has risen by 1.1°C during the period 2011-2020 when compared to the period of 1850-1900 [[101](#)]. There is a consensus that this increase of the surface temperature is caused by anthropogenic emissions stemming from human activities such as deforestation and the burning of fossil fuels and thus releasing CO_2 and NO_x to the atmosphere [[6](#)]. If the current trend of green house gas emissions continues, the Paris Agreement's [[128](#)] goal to keep the increase of the surface temperature of the globe under 2°C will not be met [[46](#)]. That means that to stop or at least to slow down the global warming, a decrease of greenhouse gas emissions is needed as much and as fast as possible. Reports published by the European Environment Agency of the European Union and the [IPCC](#) of the United Nations indicate that transportation is one of the categories with substantial CO2 emissions [[102](#)]. Cars cause over 70% of the green house gas

emissions, indicating that there is a big potential for savings. 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. This is why cities worldwide try to increase the modal share of bicycle traffic and by doing so, decrease the usage of motorized private transport. The heavy usage of motorized private transport in urban areas also cause other problems.

First, there are the motor vehicle traffic crashes with an annual death toll of over 1.3 million and an annual injury toll of over 78.2 million world wide [19]. Additionally, these crashes can cause long-lasting psychological traumas for everyone involved, such as witnesses or police officers, health workers and firefighters that have to document the crash, help injured people and rescue people who are stuck in the car wreck.

Second, there are health issues that are indirectly caused by motorized private transport. Transportation is one of the main causes for air pollution in urban areas in Europe [52] and it is estimated, that around 8 million people die due to health issues that are connected to air pollution every year [55]. Motorized private transport causes air pollution in various ways. Cars with an internal combustion engine produce (green house) gases such as CO₂ or NO_x as a byproduct from burning fossil fuels and particulate matter from the friction of the brakes and other car parts, as well as the friction between the tires and the street.

Third, noise pollution, to which car traffic undoubtedly also contributes significantly, also causes serious health issues [94]. Sleep and stress disorders, adverse reproductive outcomes are just some of the problems related to traffic noise.

Other problems of motorized private transport in urban areas and car-centric city planning are urban sprawl, which leads to less walkable cities [129] and thus a more sedentary lifestyle, which in turn causes health issues stemming from a lack of physical activities, soil sealing, which damages biodiversity and the ecosystem [162] and many more [25, 115, 136].

1.1 Problem Statement

Since cities were developed with the motorized private 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 [85], studies have shown, that the perceived safety of commuting with a bicycle is very low and one of the main reasons why people prefer other transportation modes to cycling [74]. This poses a challenge to people who want to cycle, city planners, and politicians alike. People who want to cycle have to overcome their (perceived) lack of safety and comfort while commuting by bicycle to not feel forced to travel by car. 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 and hinders them to change the traffic infrastructure in such a way that would benefit cyclists. 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.

To enable city planners and politicians to do evidence-based decision-making, they need a) to be aware of the problems in bicycle traffic, b) that a lot of people are interested in an improved bicycle traffic safety and c) to know where and how to improve the bicycle infrastructure. Cyclists need good bicycle infrastructure that increase both safety and comfort of cycling, so that more and more people choose the bicycle over other modes of transport. If the existing bicycle infrastructure is not good enough, they need to know which street sections are dangerous so that they can avoid them or be extra vigilant, when they have to pass through them. Additionally, it would motivate cyclists to know that the problem is worked on and to keep the demand for a good bicycle infrastructure high.

1.2 Contributions of this Thesis

In this thesis we addresses the problems mentioned above by fulfilling three needs: First, we conceptualize and implement a cycling trip data gathering platform, that also collects information about near miss incidents. Second, we derive the surface quality of bicycle roads. Third, we improve the bicycle simulation of a widely used traffic simulation software.

1.2.1 SimRa: A Platform for Gathering Cycling Trip Data and Detecting Near Miss Incidents in Bicycle Traffic

The SimRa¹ platform comprises all things related to the collection, storage, and analysis of crowd-sourced cycling data. The data acquisition relies on a free app, which is available on both major smartphone operating systems Android and iOS, installed on the smartphones of participating cyclists. This app collects data and detects near miss incidents² during bicycle trips, lets users add comments or labels, and anonymizes the data before uploading it to the SimRa servers. Initially, the detection of incidents used a heuristic, where the accelerometer data of the recorded bicycle trip were analyzed and the biggest differences in consecutive values were considered a incident. We then built a second incident detection option that relies on deep learning model. The anonymized data comprises information on cyclist routes, incidents, user demographics, as well as some aggregated bicycle trip statistics. Finally the collected data gets processed and analyzed continuously to gain insights into dangerous street segments and intersections. For this, we have developed one approach for interactive exploratory data analysis based on a web application and one for confirmatory data analysis which automatically derives a “dangerousness” score per street segment and intersection.

We have published this contribution in [88, 92].

¹SimRa is a German acronym for “Sicherheit im Radverkehr”, which translates to Safety in Bicycle Traffic.

²In the remainder of this thesis, we will also refer to them as “incidents”.

1.2.2 Deriving Road Surface Quality from Cycling Trip Data

Deriving road surface quality is one way how we leverage the cycling trip data gathered with SimRa. The inclusion of the accelerometer sensor readings into the bicycle trip data recordings was mainly to be able to detect incidents based on that data, since the main idea behind that is, that incidents trigger sudden movements of the cyclists and thus can be recognized in the motion sensor readings. However, a byproduct of this approach is, that the SimRa dataset also contains information about the surface quality of the roads the bicycle trip was recorded on. While the automatic detection of incidents views the vibrations stemming from the uneven surface of the roads as noise data, here we can use them to derive the “bumpiness” of the road. One major challenge, though, is the heterogeneity of the user base who contribute to the SimRa dataset. Different bicycle types and the place of the smartphone during the bicycle trip have both a big impact on how intensive the vibrations caused by the road surface are detected by the accelerometer sensors. The bicycle trip data contain information about the bicycle type and the position of the smartphone during the bicycle trip, however, these information can be given voluntarily by the user. Heterogeneous cyclists means also different styles of cycling. While some may cycle rather quiet and relaxed, others may tend to cycle fast and sporty, which can also influence the resulting motion sensor readings. Additionally, a big variety of different smartphone models result in many different hardware and software, when it comes to reading motion sensor data. The aforementioned aspects can result in different “bumpiness” values, from many bicycle trips that go through the same road. This is why our approach considers the relative road surface quality that makes it possible to successfully compare different streets with each other.

We have published this contribution in [89].

1.2.3 Cyclist Model for SUMO

The SimRa dataset contains the GPS trace of the bicycle trip as well as the timestamp for each tracked GPS point. The main reasoning to include that data is to be able to show the recorded trip to the user and to place incidents at the correct position. Additionally, by knowing where the

trip has been through, we can identify dangerous and safe areas in bicycle traffic. However, we can also extract the longitudinal movement profile, that is the acceleration, deceleration and maximum velocity of cyclists, as well as their behavior at intersections. With this, we improve the bicycle simulation model of SUMO [111], which simulates bicycles as slower cars or faster pedestrians by default, which is obviously far from being realistic. To do so, we first separate the bicycle trips into slow, medium and fast trips to distinguish between slow, medium and fast cyclists. A comparison of the longitudinal movement and the left-turn behavior at intersections shows how unrealistic the default bicycle simulation of SUMO is. Based on the three groups of bicycle trips, we each create a model for slow, medium and fast cyclists in SUMO and show the improvement compared to the default bicycle model of SUMO.

We have published this contribution in [90, 91].

1.3 Structure of this Thesis

This thesis consists of three parts. Part I: Foundations begins with this chapter and proceeds with background information on the topics of and an overview of the related work that are needed for a better understanding of the rest of this thesis.

Chapter 2

Background and Related Work

The main contributions of this thesis revolve around the SimRa project (see section 3.2.1), which is a crowdsourcing/crowdsensing citizen science application, with the goal of revealing hazardous hotspots in bicycle traffic, that are detrimental to cycling safety and comfort. In this chapter, we go through fundamental topics that are needed to better understand the nature of the SimRa project and thus, the contributions outlined in part II. For this, we first outline which factors mainly influence cycling comfort and give an overview of recent studies from that research area in section 2.1. We then continue with cycling safety in section 2.2, which is related to cycling comfort. There, we also differentiate between unilateral accidents and traffic crashes. In section 2.3, we point out similarities and differences between crowdsourcing and crowdsensing, so that we can later, in section 3.2, better categorize the SimRa project between them. Lastly, to complete this chapter, we take a brief look into the field of citizen science, namely what makes a project a citizen science project, which benefits and challenges it brings and how the related work makes use of it in section 2.4.

2.1 Cycling Comfort

According to surveys, a low level of cycling comfort is one of the main reasons why people choose other modes of transportation, especially driving a car, over cycling. This issue underlines the importance of cycling comfort when trying to increase the modal share of cycling. The Cambridge Dictionary¹ defines comfort as “a pleasant feeling of being relaxed and free from pain”.

According to that definition, a comfortable ride needs to be absent of both *mentally* and *physically* stress factors. Related work has identified four factors that can cause mental and physical strain on the cyclist: Cycling Posture, Cycling Smoothness, Environment, Mental State.

Cycling Posture

A wrong cycling posture does not only result in a lower cycling comfort, it can also cause medical problems, especially over long periods of times, such as painful pelvic bones or pain in the region of the perineum [40]. Causes for a wrong cycling posture can be a wrong-sized bicycle frame, a misaligned handlebar or a saddle that is too high, too low or badly angled. The shape of the saddle as well as the handlebar are also important, as different types of saddle and handlebars result in different cycling postures [27]. Informing cyclists about the importance of cycling posture, so that they can buy the right bicycle for them, as well as adjust the bicycle they already have, seems to be a good way to improve the cycling postures of cyclists and thus increasing their cycling comfort.

Cycling Smoothness

Vibrations stemming from the rolling of the bicycle wheels on the surface can also lead to a decrease of comfort level. These vibrations are mainly conducted to the cyclist through the contact points with the bicycle, usually at the handlebar and the saddle. This means, that if the handlebar and the saddle absorb the vibrations of the ride, the cycling will be more smooth and thus more comfortable for the cyclist. Soft materials or good suspensions, as well as good, vibration-absorbing

¹<https://dictionary.cambridge.org/>

tires can contribute here. But the biggest factor here is obviously the road surface. Without a well-built and well-maintained infrastructure, the best bicycle in terms of comfort cannot absorb the vibrations of the unevenness, the cracks and the potholes of a badly maintained bicycle lane or road.

Environment

The environment is a significant factor that influences the comfort level of a cycling trip [76]. Thereby, we can distinguish between the *natural* and the *built* environment. To the *natural* environment belong climate, weather, scenery and topography. Most cyclists find a bad weather, i.e., strong winds, precipitation too low and even too high temperature combined with humidity [124], detrimental to their cycling comfort. The study by Boyce et al. [24] shows that people feel more comfortable during daytime, so long winters with short day times, as is common near the poles, also leads to less comfortable cycling trips. Wessel et al. [180] look further into this issue and find out, that adopting the day saving time throughout the year would increase the amount of cycling. A nice scenery along the cycling trip route can increase cyclist satisfaction and thus result in a higher comfort level [174, 181]. While hills can be advantageous for recreational cycling trips due to a nice scenery and the physical effort to cycle up to a hill, that is commonly sought after by recreational cyclists, they are often perceived as a nuisance by commuting cyclists [100].

So the *natural* environment can impact the comfort of a cycling trip both in positive and negative ways. Similarly, the *built* environment can further enhance the positive and negative effects of the *natural* environments. As the name suggests, *built* environment refers to environmental circumstances realized by humans. The most obvious one to mention here is the bicycle infrastructure. Not only does a well planned and maintained bicycle infrastructure reduce unpleasant vibrations and bumps during the cycling trip, it also contributes to a comfortable ride by making clear how the cyclist has to behave in traffic, e.g., when approaching an intersection. A well maintained infrastructure is also important to mitigate the downsides of bad weather, especially the removal of accumulated snow on bike lanes/path is crucial [4, 152]. Likewise, street lights illuminating the

bicycle infrastructure leads to a more comfy cycling trip, because it makes it easier for the cyclist to perceive the surroundings and avoid potential hazards [47].

In general, the physical well-being and a feeling of safety is very important for comfortably cycling, which is why the next section deals with cycling safety.

2.2 Cycling Safety

A (perceived) lack of safety is one of the main factors why potential cyclists prefer other modes of transportation over cycling [54, 99, 122]. Hence, it is important to understand what increases or decreases the (perceived) safety of cycling and how to make cycling more safe, both objectively and subjectively. The emphasis on perceived/subjective safety is important in two ways: On one hand, if the perceived safety of a road segment is higher than the objective safety, cyclists are not aware of the risk they put themselves in [39]. On the other hand, when the safety of cycling is perceived as to be lower than it actually is, potential cyclists stay away from cycling due to a misconception [125].

First, we look into the objective safety in bicycle traffic, before we highlight the issues regarding perceived safety. The main safety concern regarding cycling safety are accidents. These can be differentiated into two categories: unilateral accidents, such as falling from the bike due to losing balance or slipping in winter conditions and traffic crashes with other participants,.

Unilateral Accidents

In their studies, Juhra et al. and Naess et al. investigate bicycle accidents, that were not reported to the police by analyzing hospitalization reports, where the patient was admitted after a bicycle accident [85, 121]. This is a good way to study unilateral accidents, since these are usually not reported to the police. Unilateral accidents make up the majority of bicycle accidents [85, 121] and the most common form is falling from the bicycle without external forces. Reportedly, most of these accidents happen to young and middle aged adults during the rush hours, which correlates

with an increased bicycle traffic activity during these times. Interestingly, children, probably due to their inexperience, and the elderly, probably due to the tolls of the age on the human body, experience much more unilateral accidents than traffic crashes with other participants. Better bicycle infrastructure, i.e. well maintained and well lit bicycle roads, especially during winter, are needed to decrease the risk of falling from the bicycle without external forces, since they likely happen due to bumps in the road that were seen too late to circumvent by the cyclists. Another interesting finding is, that during night times the cyclists, that fall from the bicycle without external forces are young adults, probably returning from partying, thus raising awareness could also decrease this type of accidents.

Other types of unilateral accidents are collisions with fixed objects, such as trees, or accidents caused by a technical defect of the bicycle. Statistics about collisions with fixed objects are very similar to falling from the bicycle without external forces, which further supports the above mentioned causes and proposed solutions. Especially younger people such as students tend to have cheaper bicycles due to their lower income and fear of their bicycle getting stolen. Cheap and old bicycles are more susceptible towards technical defects, which is probably why younger people have more accidents caused by them. More awareness and easier access to bicycle repair stations could mitigate this problem.

Traffic Crashes

The predominant form of traffic crashes in bicycle traffic are accidents with motorized vehicles [85, 121]. Similar to unilateral accidents, traffic crashes mostly happen to young and middle-aged adults during the rush hours. This is no surprise, since these are also the times with the biggest motorized vehicle traffic volumes, which lead to a bigger exposure of cyclists to motorized vehicle traffic. In contrast to unilateral accidents, traffic crashes happen more during daytime. The fact that there is more traffic during daytime in general and that motorized vehicle drivers drive more carefully, when the view is impaired, e.g., by darkness or bad light conditions [149]. It is also worth noting, that traffic crashes usually result in more serious injuries and mortality than unilateral accidents. These findings stress the necessity for a separation of bicycle and motorized vehicle traffic with

physically separated bicycle tracks, since these reduce the risk of a crash between them by at least 50% [131].

2.3 Crowdsourcing and Crowsensing

Crowdsourcing and crowsensing have emerged as successful approaches to solve tasks in a fast and cost-efficient way and gather big amounts of data in a short time respectively.

Crowdsourcing

“Crowdsourcing” is a combination of the words *crowd* and *outsourcing* [75]. *Crowd* refers to an often larger group of individuals, to which a task is transferred to, whereas outsourcing refers to a practice where an entity does not fulfill a task in-house on its own, but hires people from outside to solve a task for them. Crowdsourcing comes in many shapes and forms and can be found in many different areas, from transportation, to financing, from social media to natural crisis relief, etc. A very common form of crowdsourcing is the so called mobile crowdsourcing. The difference is that in mobile crowdsourcing the people fulfilling the task usually do not have to be stationary, but rather mobile, because the task at hand demands movement or being in a specific location to be fulfilled [133]. In the latter case, a smartphone or a hardware device fitted with sensors has to be used by the participants to fulfill the task. For the remainder of this work, we will use the terms crowdsourcing and mobile crowdsourcing interchangeably.

Figure 2.3 shows the key elements of crowdsourcing. In the top layer, the application user (e.g., a ride-hailing app user) requests a task (e.g., a trip from location A to location B). The data processing platform layer receives this task and determines to whom (e.g., which taxi drivers) to allocate it to. Then, the crowd (e.g., the taxi driver, that accepts the task) carries it out. After that, the data processing layer receives the result (e.g., the arrival of the customer at the destination), processes and aggregates it and sends the end result to the application (e.g., invoice). In crowdsourcing, it is also possible and sometimes even necessary that the crowd has to

collaborate to fulfill the task (e.g., multiple taxi drivers transporting a large group of customers from A to B).

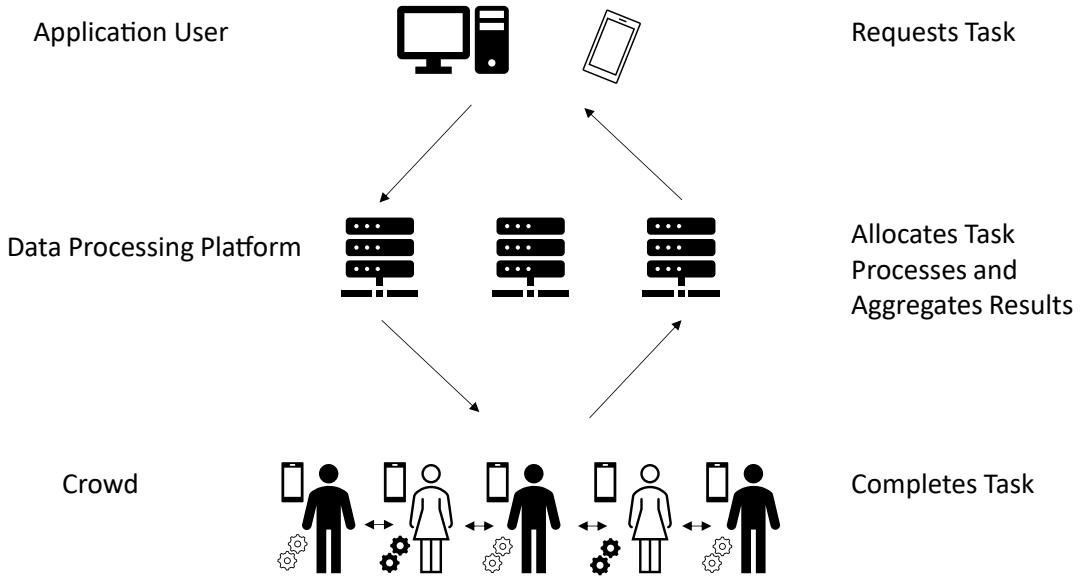


Figure 2.1: Crowdsourcing architecture consist of three layers: Application, Data Processing and Crowd

There are different aspects that characterize a crowdsourcing application: *degree of human involvement*, *location relevance*, *knowledge requirement*, *participation incentive* and *data flow* [95, 141].

There are two types of *degree of human involvement*: Opportunistic and Participatory. The former describes that the participant does not actively operate the smartphone or device, while in the latter, he/she has to give further input on the smartphone or device. The *degree of human involvement* is a continuum, where each crowdsourcing application can have different levels of human involvement.

Location relevance refers to the importance of solving a task in a particular location. With mobile crowdsourcing, a certain degree of *location relevance* is naturally given, but there can still be different levels of *location relevance*. For instance, the task may need to be completely done at a specific locations, or only parts of it, such as taking a photo of a location but then editing the photo anywhere. Similar to the *degree of human involvement*, here we also have a continuum, being between high location relevance and no location relevance at all.

Knowledge requirement refers to the level of knowledge that the participant needs to fulfill a given task. Some tasks may be very simple, where detailed knowledge is unnecessary, while other tasks may need some experts of a specific topic or an extensive training to be able to solve the task. A high *knowledge requirement* may make it necessary to distinctively allocate the task to such participants, where the *knowledge requirements* are met.

The *participation incentive* can vary from project to project and also from participant to participant. There are basically two types of incentives: intrinsic and extrinsic. A participant with an intrinsic incentive engages with a crowdsourcing project, because he/she can identify with the problem the project wants to solve or simply likes to perform the task. There can also be extrinsic motivations such as getting paid in cash or with coupons, or gaining social credits for it. It is very much possible, that participants have both intrinsic *and* extrinsic incentives to contribute a project.

The *data flow* is a technical aspect of the crowdsourcing project. Here, there are three distinctions: centralized, decentralized and hybrid. When the data flows from the crowd directly to the processing platform, the project has a centralized data flow. Contrary to that, when the data flow is decentralized, it means that some participants' devices collect results of the solved tasks within the crowd and then send it in aggregated form to the processing platform. This may be beneficiary to decrease the load on the processing platform. A hybrid form is also possible, where the crowd or the crowd's devices process the data themselves and send it directly to the application, which makes the processing layer superfluous.

Crowdsensing

As in “crowdsourcing” the crowd in “crowdsensing” refers to a group of people, who voluntarily take over a task. However, in *crowdsensing*, the task at hand is mainly gathering data for the client, or in other words, *sensing*. Due to the similar nature of both, the architecture of crowdsensing projects is usually very similar to that of crowdsourcing projects (see figure 2.3). The main difference from a software architectural view is that the participants usually do not collaborate

and thus there is no need for the smartphones and devices in the crowd to communicate with each other.

The characterization is also very similar, although there is usually no *knowledge requirement* aspect in *crowdsensing* projects. However, there are two additional aspects, that can be used to characterize: *application type* and *data collection*.

There are mainly three *application types*: environmental applications, infrastructure applications and social applications [56].

In *environmental applications*, the goal of the crowdsensing project is to gather data about the *natural environment*. Examples of *environmental applications* are projects to measure air pollution [66, 110, 154] or monitor water quality [119, 140, 151].

infrastructure applications are about gaining insights into the state of public infrastructure. Common use cases are gathering data about car parking spaces [42, 142, 173] or traffic flow [104, 116, 176].

As the name suggests, *social applications* gather information about persons and social interactions. With the help of social media networks, it is possible to extract information from the communications of the crowd [31, 61, 132], while there is also a focus on mass events [30, 83, 138].

The *data collection* can come in two forms: *mobile sensing* and *user generated* [134].

In *mobile sensing* the smartphone or sensing device collects the data automatically. The participant may need to start an app or turn on a device to start a recording of the sensor data. When the smartphone or device that collects the data sends the collected data to the data processing platform autonomously, the data collection is *push based*, otherwise, if the data processing platform requests the collected data from an endpoint it is called *pull based*.

2.4 Citizen Science

Although the term Citizen Science was coined by Irwin [79] and Bonney [23] in the mid-90s, the concept is much older. For example, the National Audubon Society’s project called “Christmas Bird Count”, where everyone is invited to participate to count birds and report their findings. It runs since the year 1900 and can be considered as the first Citizen Science project [153]. This can be considered as Citizen Science because it is a scientific endeavor, where anyone can participate, which is the broadest way to describe Citizen Science. However, since the emergence of the term “Citizen Science” different definitions have been proposed by different scientific, social and political parties [7, 70, 169] and there is an active discussion of what can be considered as a Citizen Science Project and what cannot [65].

Haklay et al. compiled a list of 32 definitions for Citizen Science from different reference sources, citizen science associations, global multinational organizations and different governmental bodies [65]. One of their key findings that although most definitions have descriptive, instrumental and normative elements, the weighting can differ according to the goals of the proposing party of the definition. *National Geographic* has for example a more descriptive definition:

Citizen science is the practice of public participation and collaboration in scientific research to increase scientific knowledge. Through citizen science, people share and contribute to data monitoring and collection programs.

After a broad description of what Citizen Science is, it underlines “data monitoring and collection” which is not surprising for a environmental-focused organization.

The *US National Institutes of Health* chooses a more instrumentalist approach when defining Citizen Science:

Citizen science efforts are driven by community concerns. These community-led projects may involve a partnership with an academic or research institution, where both parties work together to collect and share data. The goal is to address a community concern

through collaborative research and to translate the research findings into public health action that benefits the community.

The last sentence of their definition clearly puts a focus on public health, which is not understandable from their point of view. There are also definitions who try to be normative, such as the definition from UNESCO:

[Citizen Science:] The participation of a range of non-scientific stakeholders in the scientific process. At its most inclusive and most innovative, citizen science involves citizen volunteers as partners in the entire scientific process, including determining research themes, questions, methodologies, and means of disseminating results.

It is remarkable that this definition mentions inclusiveness and voluntarism.

Eitzel et al. [49] also analyze different definitions of Citizen Science and identify three categories: Citizen Science as a *tool*, a *movement* or a *social capacity*. Accordingly, Citizen Science can be seen as a *tool* to help answering a research question, e.g. by gathering data. On the other hand, Citizen Science can also be seen as a *movement* to democratize science and thus increase the trust of the society in science. Finally, Citizen Science can be seen as a *social capacity* to increase evidence-based decision making.

The European Citizen Science Association (ECSA)² contributes the ten principles of citizen science [7]:

1. *Citizen science projects actively involve citizens in scientific endeavour that generates new knowledge or understanding. Citizens may act as contributors, collaborators, or as project leader and have a meaningful role in the project.*
2. *Citizen science projects have a genuine science outcome.* For example, answering a research question or informing conservation action, management decisions or environmental policy.

²<https://www.ecsa.ngo/>

3. *Both the professional scientists and the citizen scientists benefit from taking part.* Benefits may include the publication of research outputs, learning opportunities, personal enjoyment, social benefits, satisfaction through contributing to scientific evidence e.g. to address local, national and international issues, and through that, the potential to influence policy.
4. *Citizen scientists may, if they wish, participate in multiple stages of the scientific process.* This may include developing the research question, designing the method, gathering and analysing data, and communicating the results.
5. *Citizen scientists receive feedback from the project.* For example, how their data are being used and what the research, policy or societal outcomes are.
6. *Citizen science is considered a research approach like any other, with limitations and biases that should be considered and controlled for.* However unlike traditional research approaches, citizen science provides opportunity for greater public engagement and democratisation of science.
7. *Citizen science project data and meta-data are made publicly available and where possible, results are published in an open access format.* Data sharing may occur during or after the project, unless there are security or privacy concerns that prevent this.
8. *Citizen scientists are acknowledged in project results and publications.*
9. *Citizen science programmes are evaluated for their scientific output, data quality, participant experience and wider societal or policy impact.*
10. *The leaders of citizen science projects take into consideration legal and ethical issues surrounding copyright, intellectual property, data sharing agreements, confidentiality, attribution, and the environmental impact of any activities.*

The plethora of different definitions, guidelines and principles show that Citizen Science is a broad term that can be interpreted according to the circumstances. This, however, is also a benefit, since it allows different projects to label themselves as Citizen Science and get funding from different organizations and national institutions.

Part II

Improving Safety in Bicycle Traffic

In this part, we exhibit the three main contributions of our work. First, we introduce SimRa, which is a mobile application for gathering cycling trip data and we also focus on our approach to automatically detect incidents in bicycle traffic in chapter 3. Second, we suggest an approach to derive the road surface quality from cycling trip data and use the data gathered with SimRa to demonstrate how it works in chapter 4. Third, we show our approach to improve the bicycle simulation of an urban traffic simulation software called SUMO, by utilizing the SimRa dataset to extract the lateral movement and the left-turn behavior of cyclists at intersections in chapter 5.

Chapter 3

Gathering Cycling Trip Data and Detecting Incidents

In this chapter, we present SimRa¹, which is a crowdsourcing (see section 2.3) platform for gathering cycling trip data. A special feature of SimRa is its capability to automatically detect incidents in bicycle traffic, which we also show in this chapter. SimRa's main objective is to find out hazardous hotspots in bicycle traffic. For this, we need to gather recordings of cycling trips, which have to contain user-generated information regarding the incidents, as well as a GPS trace, so that we can map the incidents and find out where they accumulate.

However, a cycling trip can be very long, both in distance and time, which complicates this process for the SimRa users since it can be difficult to remember every incident, its details and exact location. While automatically recording the GPS trace during a cycling trip via a smartphone app is trivial, automatically gathering detailed information such as exact location, type and other participants of the incidents is not. It is however possible for the SimRa app to make educated guesses *where* an incident might have happened during a ride based on the data produced by the Inertial Measurement Unit ([IMU](#)), which provides us information about acceleration, orientation and other gravitational forces. With this information, the user only has to remember whether an incident really happened there and what kind of incident it was, which makes their task easier. For

¹SimRa is an acronym for **Sicherheit im Radverkehr**, which translates to safety in bicycle traffic from German.

this, SimRa has two different mechanisms. The first mechanism relies only on the [IMU](#) readings of the ride to suggest points in the ride where an incident might have happened and was available since the initial release of SimRa. The second mechanism relies on a Deep Learning ([DL](#)) model which was trained with labeled data and is better at detecting incidents, which eases the task of the SimRa users to provide all incidents together with their details.

This chapter combines material published in [88, 92] and has following contributions:

- We describe the design of the SimRa platform, a crowdsourcing-based data collection and processing platform for cyclist routes and incidents.
- We propose an approach that combines signal processing and Machine Learning ([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.
- 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.
- We discuss to which degree our approach can automate incident detection and which additional sensors are needed for full automation.

This chapter continues with background information about [DL](#) and Time Series Classification ([TSC](#)) following Ismail Fawaz et al. [80] in section 3.1, which is important to understand the second mechanism to automatically detect incidents, mentioned above. Afterwards, we give a more detailed description about the SimRa mobile application in section 3.2. In section 3.3, we explain how we use [DL](#) to automatically detect incidents. We evaluate our approach in section 3.4, which we follow up with a discussion in section 3.5. Finally, we give an overview of alternative approaches and summarize this chapter in section 3.6 and section 3.7 respectively.

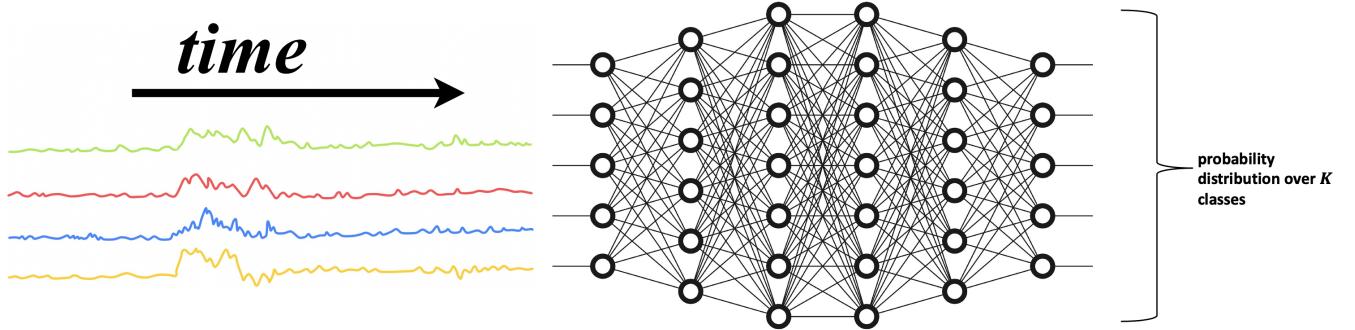


Figure 3.1: Structure of a Deep Learning approach to time series classification: A classifier that maps a time series to a probability distribution of classes[35]

3.1 Deep Learning for Time Series Classification

In this section, we explain background information about time series classification and deep learning, which is used by SimRa to automatically detect the incidents.

3.1.1 Time Series Classification

As we will see in section 3.2, the SimRa dataset consists of multivariate time series. A multivariate time series consists of $n \in \mathbb{N}$ different univariate time series $X_i \in \mathbb{R}_t$ with univariate time series $X = x_1, x_2, \dots, x_t$ with each $x_i \in \mathbb{R}$ being chronologically ordered by $t \in \mathbb{N}$. In short, a multivariate time series is a chronologically ordered set of variables, in our case different IMU readings. An example of a univariate time series is the data produced by a distance sensor, where we have pairs of $(\text{distance}, \text{timestamp})$, chronologically ordered by timestamp . An example of a multivariate time series is a weather station database, that reports the temperature, humidity and wind force every hour. The task of TSC is to assign a one-hot label to each entry of the time series, e.g., whether the weather is safe for an airplane to fly or not.

3.1.2 Deep Learning for Time Series Classification

With their rise, Deep Neural Networks (DNNs) have drawn attraction to themselves from the field of TSC.

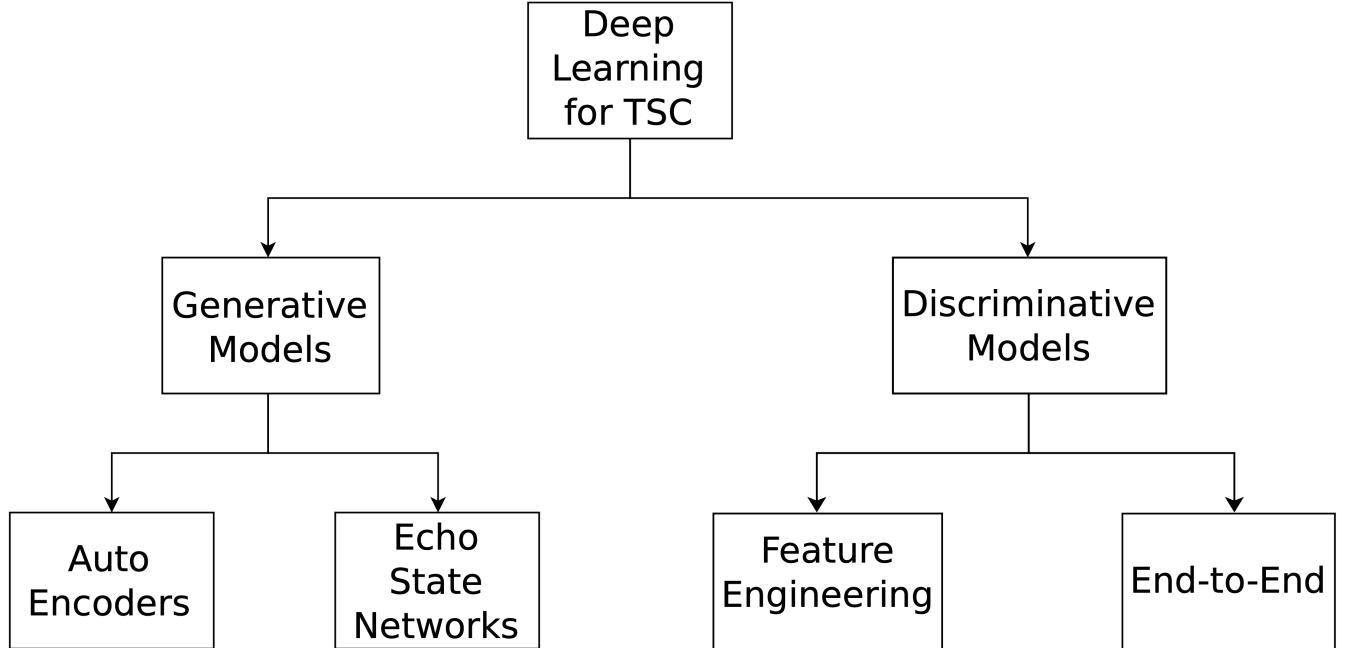


Figure 3.2: Taxonomy of Deep Learning approaches for Time Series Classification (based on Länkvist et al. and Ismail Fawaz et al. [80, 98])

As shown in figure 3.1, a Deep Learning approach to [TSC](#) consists of a classifier (in this case a Multi-Layer Perceptron ([MLP](#)), which is a type of a [DNN](#)), that maps a time series to a probability distribution of K classes.

Figure 3.2 shows the two categories of [DL](#) approaches for [TSC](#), which were differentiated by Länkvist et al. and then extended by Ismail Fawaz et al. [80, 98].

Generative Models

Generally, generative Models provide a probability distribution $P(x, y)$ with x being the input data and y being the label. The label y is determined by using the Bayes rules to calculate $P(y|x)$ [123]. A generative model for [TSC](#) usually consists of two parts [98]: First, a model of the dataset is derived with an unsupervised learning phase. This model is then used to train a classifier that can successfully determine the correct label for each input. Ismail Fawaz et al. classifies two different categories of Generative Models: *Auto Encoders* and *Echo State Networks* [80].

Auto Encoders, such as Stacked Denoising Auto-Encoders [15], Convolutional Neural Networks (CNNs) [155], Deep Belief Networks [10], Recurrent Neural Networks (RNNs) [139], split the the unsupervised learning preprocessing and the training of the classifier.

Echo State Networks on the other hand combine these steps [82] to first reconstruct the dataset, to use the learned representation of it from the reservoir space [8, 20, 37, 113]

Distinctive Models

As stated by Vapnik [172], the classification problem should be solved directly, without trying to solve a more general problem as an intermediate step. This is what Distinctive Models do, and instead of first calculating $P(x, y)$, they derive $P(y|x)$, or a mapping from an input to the label directly [123]. Distinctive Models can be further categorized into *Feature Engineering* and *End-to-End* [80].

As the name suggests, in *Feature Engineering* the focus lies in the hand-engineering of the features. In the most common form, which was inspired from computer vision, the time series is first transformed into an image [67, 165, 177], so that the features could be hand-engineered correctly. This approach does not need any domain-specific knowledge. If domain-specific knowledge is available, the feature extraction can be done without transforming the dataset into images and the features can directly be included in the classifier [77, 168].

Contrary, in *End-to-End* Deep Learning the feature extraction is included in the process and does not have to be made manually [126]. Distinctive Models of the *End-to-End*-type have been realized with MLPs [58, 178], CNNs [34, 81, 109] and hybrid networks [107, 150].

3.2 SimRa - A Platform for Gathering Cycling Trips and Incidents in Bicycle Traffic

In this section, we give a general overview of the SimRa platform which comprises all things related to the collection, storage, and analysis of crowdsourced cycling data which we will focus on in the following sections.

3.2.1 Goal of the SimRa platform

For data acquisition we rely on an app installed on the smartphones of participating cyclists. This app collects data and detects incidents during bicycle trips, lets users add comments or labels, and anonymizes the data before uploading it to our servers (see section 3.2.2). The anonymized data comprises information on cyclist routes, incidents, user demographics, as well as some aggregated trip statistics. Finally, we continuously process and analyze collected data to gain insights into dangerous street segments and intersections. For this, we have developed one approach for interactive exploratory data analysis based on a web application and one for confirmatory data analysis [18] which automatically derives a “dangerousness” score per street segment and intersection ².

3.2.2 Data Acquisition

For data acquisition, we could either rely on dedicated hardware or use commonly available hardware such as smartphones. While dedicated hardware has certain benefits, e.g., higher measurement precision, such projects are inherently limited in scale: An example of this is the Radmess project ³ which built custom sensors to track close pass incidents. While the data quality is indeed rather high, the project was limited to a total of 100 cyclists (only partly in parallel). As a result, the project could not achieve full coverage of Berlin streets. For example, some incident hotspots in terms of close passes that we were able to identify in SimRa did not even have a single bicycle

²<https://simra-project.github.io/dashboard/>

³<https://interaktiv.tagesspiegel.de/radmess/>

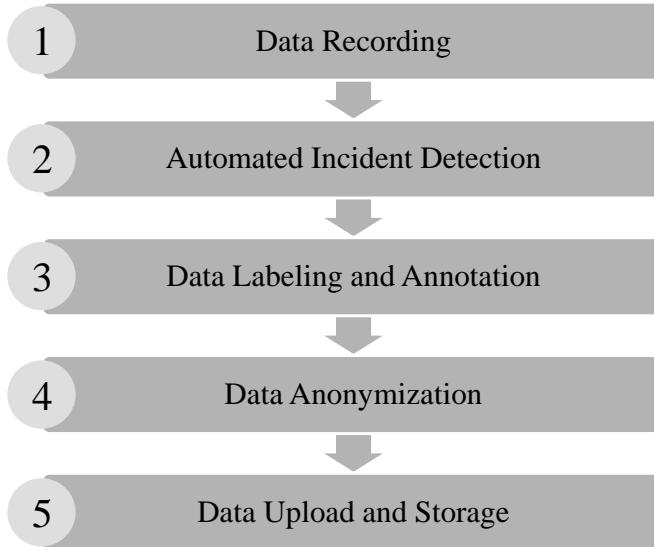


Figure 3.3: In the data acquisition process, collected data is manually annotated and anonymized before being uploaded to our backend servers.

trip in the Radmesser data. We decided for project scalability and collect data with smartphones only. Our goal is to collect data in a way that allows us (i) to identify incident hotspots as well as the kind of incidents and (ii) to identify the routes of cyclists (in which unnecessary detours are likely to identify severe incident hotspots).

Overall, our data acquisition process has five steps and follows the structure shown in figure 3.3 (we describe the steps in the following in detail). This process runs continuously and in parallel as cyclists may create data for individual bicycle trips at any time. During a trip, we first record sensor data using the built-in sensors of a cyclist’s smartphone. Upon completion of a trip, we analyze the raw data to automatically detect incidents. Afterwards, the cyclist can enrich collected data with labels and annotations, use a number of anonymization measures, and upload the data to our backend servers.

Data Recording In the SimRa app, users manually start recording before they begin cycling (left part of Figure 3.4). During a cycling trip, we track three sensors at varying rates per minute. First, we query the GPS sensor every three seconds; this returns the current location and a radius with an accuracy confidence value of 68%. Second, we query the smartphone’s accelerometer at 50Hz. While such a high sampling rate allows us to detect sudden peaks, this also leads to an

unnecessary large data set which typically needs to be uploaded via mobile networks. Thus, we aggregate the data based on a moving average across 30 values of which we only consider every sixth value. This reduces the amount of data while still retaining all peaks in sensor readings. Third, we store the device orientation based on the smartphone’s gyroscope sensor every three seconds. These values describe the rate of rotation around the X/Y/Z axis in rad/s and allow us to correctly interpret the direction of the acceleration values. Each sensor measurement, together with a timestamp, is stored locally on the device.

We chose these rate settings based on initial experiments in which we identified the data collection rates and aggregation schemes as a sweet spot between system overload and information loss.

Automated Incident Detection After a cycling trip, as soon as the cyclist stops the recording, we analyze the recorded data to identify incidents. The challenge, here, is to reliably detect incidents – initially, without any training data.

For this reason, we developed a heuristic for incident detection that relies on the assumption that incidents will often materialize as sudden acceleration spikes. After reaching over 10,000 labeled trips, we started to explore alternative detection methods ranging from machine learning to signals processing. In our heuristic, we group the acceleration time series in three-second buckets to differentiate incidents and poor road conditions (e.g., potholes result in high vertical acceleration). In each bucket, we identify the minimum and maximum value for every dimension and calculate the difference between those two. In a second step, we categorize the six highest difference values across all buckets as likely incidents. This allows us to separate high acceleration values based on poor road conditions (which usually have low difference values) from incident-related peaks.

In practice, this heuristic works well for cyclists with a “relaxed” cycling style. For cyclists with a more “rapid” style of cycling, our heuristic usually identifies either accidents, severe bumps, or traffic lights but usually not incidents. The heuristic is also inherently limited as it cannot detect close passes and similar incidents which do not materialize as acceleration spikes.

There was an alternative approach [145] using a simple Fully Connected Network (FCN) architecture without considering gyroscope sensor features yet.

Data Labeling and Annotation Even though we plan to improve the automated detection of incidents, some incident types cannot be automatically detected based on the sensor data alone. For example, while being tailgated might make the cyclist increase speed, this kind of observable activity can also be related to other, non-dangerous events. Thus, we do not think that a fully automated detection, also based on our hardware limitations, is a realistic option for SimRa – neither now nor in the future. Instead, we ask the cyclist to edit the pre-detected set of incidents (i.e., add false negatives and ignore false positives) and to label and annotate the correct set of incidents (right part of Figure 3.4).

Data Anonymization One of our side goals in SimRa is to preserve the privacy of our users, which is mainly achieved through three mechanisms: *Delayed recording* allows users to define a time and a distance threshold after which a recording will start, *trip cropping* allows users to crop their bicycle trip manually to hide where they started or arrived (middle part of Figure 3.4), and *per-record pseudonymization* stores demographic and trip data separately so that they cannot be connected to individual users. Furthermore, each trip is pseudonymized separately.

Data Upload and Storage Finally, and only when explicitly triggered by the cyclist, the cycling trip data is uploaded to our backend. For authentication, we calculate an access key based on the current timestamp and a random salt which we update with new app versions. This is necessary to avoid automated attacks on our backend as we do not have a notion of user accounts. So far, this has sufficed as extracting the salt from the app binary requires enough manual effort to make this infeasible for automated attacks. Note, that we store trips and user data per region so that we can analyze (geographic) regions separately.

Resulting Data The SimRa data set consists of more than 114,500 rides from over 100 regions. Berlin accounts for the majority of rides of any region: Almost half the rides and incidents have been recorded there. Hanover and Nuremberg are the next largest regions with approximately 8000 and 5500 rides respectively⁴. The SimRa data set and source code are available online and

⁴65,000 rides from over 65 regions and 3500 rides for Hanover and Nuremberg each, when we submitted our paper [92]

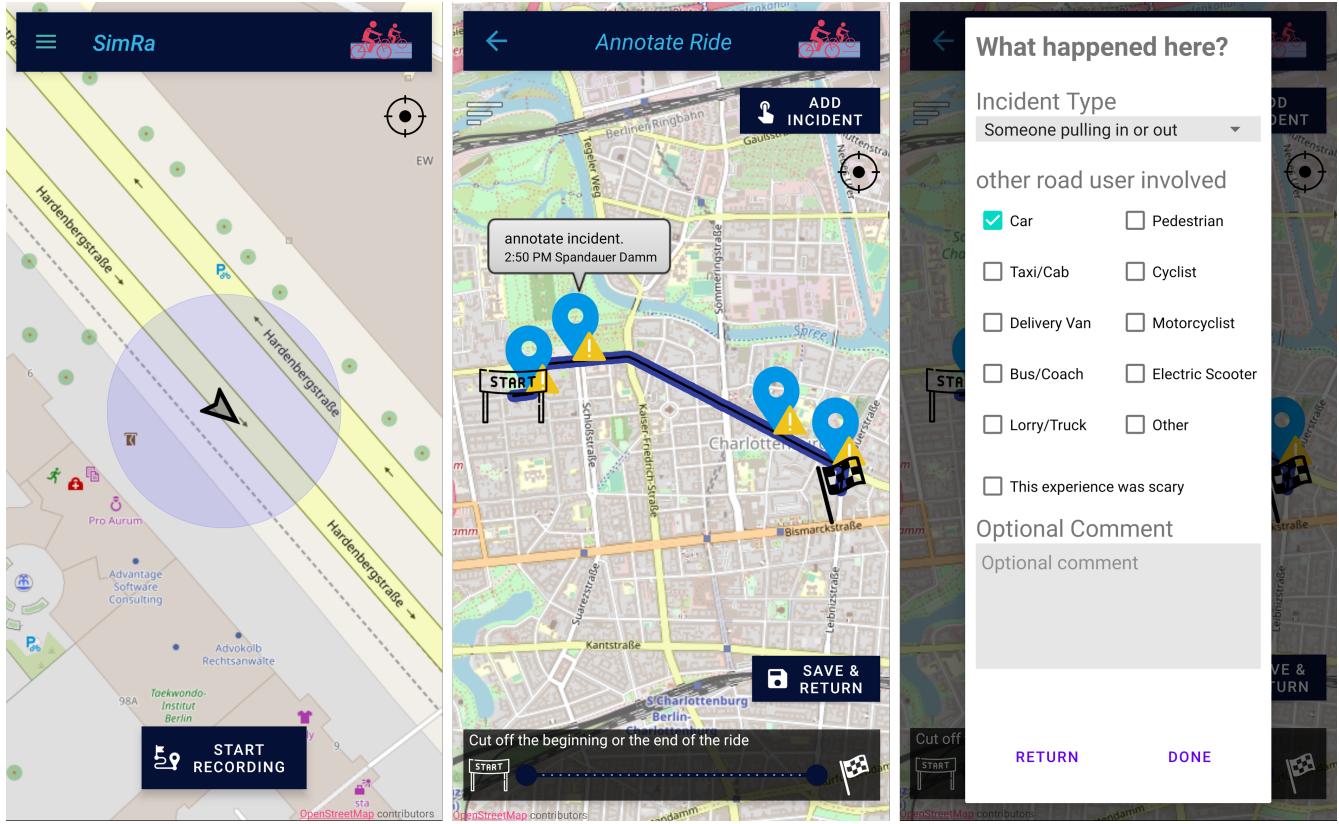


Figure 3.4: Screenshots of the SimRa app depicting the user story from left to right.

in data repositories⁵. Due to per ride pseudonymization, we used only ride data for our classifier and could not rely on profiles of individual cyclists. Each ride file has two parts: The incident part and the ride part. The incident part lists all incidents of the ride. The ride part consists of timestamps, GPS, accelerometer, and gyroscope sensor readings. In more recent versions of the app, linear accelerometer and rotation vector data are also included. The recording frequencies of the various sensors vary a lot (See table 3.1) and as a result require a partition of the SimRa data set in three distinct parts: older Android rides, newer Android rides, and iOS rides.

Crowdsensing or Crowdsourcing? Neither of the terms crowdsensing and crowdsourcing (see section 2.3) fit perfectly to SimRa. While SimRa could be classified as a *infrastructure application* with a *mobile sensing* approach as *data collection* in the context of crowdsensing, the participants have to contribute more than just the data from the *IMU*. They are also asked to classify, annotate

⁵<https://github.com/simra-project/>

	Accelerometer	Gyroscope	Global Positioning System (GPS)	Linear accelerometer	Rotation vector
Android old	10 Hz	0.33 Hz	0.33 Hz	/	/
Android new	4 Hz	4 Hz	0.33 Hz	4 Hz	4 Hz
iOS	10 Hz	0.33 Hz	0.33 Hz	/	/

Table 3.1: Theoretical measurement frequencies of different sensors in different parts of the SimRa data set. Note that these are only the theoretical frequencies that deviate significantly from the empirical measurement frequencies that can be observed in the data set (see Section 3.5).

and describe the incidents they encounter during their cycling trip. If we go through the different aspects that characterize a crowdsourcing application (see section 2.3) we can say that SimRa is a light-weight crowdsourced application. The *degree of human involvement* is medium. While the users have to start and stop the recording and have to give additional information about the ride and incidents, no input is needed during the cycling trip and the app records the trip passively. There is rather higher degree of *location relevance*. Although the annotation of the ride can be done wherever, a cycling trip can only be recorded when the cyclist is physically at a location. It is not possible to record a ride in a location without being physically there (at least not without spoofing the GPS location of the smartphone). There is a minimal *knowledge requirement* the participants have to fulfill. They must know, that they only have to report incidents and not a static hazard or a nuisance such as a very long red light. The *participation incentive* in SimRa is purely intrinsic. There are no external benefits in using SimRa, such as monetary compensation. The *data flow* is hybrid, since there is preprocessing involved in the SimRa app, that happens on the smartphone. However, it would be wrong to label SimRa as a fully fledged crowdsourcing project, since the *Application User* and *Data Processing Platform*, are basically merged together and there is no allocation of tasks. The task is to record cycling trips, add additional details about the trip and the incidents and upload the data. Hence, we consider SimRa as a (lightweight) crowdsourcing project.

SimRa and Citizen Science SimRa can clearly be considered as a Citizen Science project, as it fulfills the Ten Principles of Citizen Science (see section 2.4):

1. The users are crucial for the project, as they contribute their cycling trip data.

2. There is a genuine science outcome, as this thesis shows and the informing management decisions and environmental policies.
3. While the benefits of the citizen scientists are not as apparent as the professional scientists, they also profit in the long run, if the project goal improving the bicycle traffic safety increases.
4. While the citizen scientists' main task is to contribute cycling trip and incident data, they are free and encouraged to also participate in other stages of the scientific process. E.g., the dataset is published as open data and everyone can analyze the data for insights about traffic safety.
5. We provide multiple channels for the citizen scientists to inform themselves about the progress of the project, such as through our social media account⁶ as well as a website⁷.
6. In each of our published papers, as well as in this thesis, we point at possible limitations and biases (see sections 3.5, 4.6, 5.5 and 7.1).
7. We publish the SimRa dataset as open data [16, 17, 87]
8. While we don't mention any citizen scientists directly, since there would be too many, we make sure that it is obvious, that we rely on participating citizen scientists.
9. In each of our published papers, as well as in this thesis, we evaluate the scientific output (see sections 3.4, 4.5 and 5.4).
10. We designed SimRa to be as privacy friendly as possible and have a very strict privacy policy statement⁸.

⁶https://twitter.com/SimRa_App

⁷<https://simra-project.github.io/>

⁸<https://www.tu.berlin/en/mcc/research/projects/simra-privacy-policy-statement>

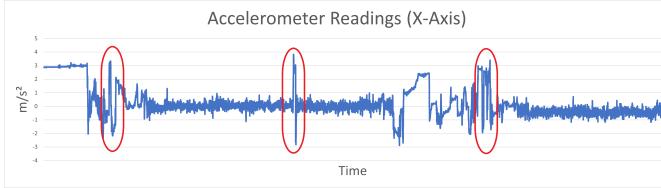


Figure 3.5: The accelerometer readings of an example bicycle trip. The encircled spots could indicate an incident but also driving over a curb.

3.3 Detecting Incidents with Deep Learning

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 3.3.1). We describe our ML model in Section 3.3.2 and the training process in Section 3.3.3.

3.3.1 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 bicycle trips.

Before the preprocessing phase, a typical trip 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 trip $i \in \{1, \dots, R\}$.

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 trips. The non-sensory features such as phone 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 prac-

tically impossible for a human. When visualizing the accelerometer sensor readings, incidents usually stand out as sudden spikes (see Figure 3.5), 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 trips 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. [166] 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 [170, 188]. 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 [179]. 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** [48, 68], 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 [22].

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 oc-

curred. For that reason, we split our ride data into 10-second buckets, following a non-overlapping sliding window approach [127]. 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 Discrete Fourier Transform (**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 [35].

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.

3.3.2 Model Architecture

As our problem setting is similar to the Human Activity Recognition (**HAR**) task (see Section 3.6), we build a customized Artificial Neural Network (**ANN**) inspired by the DeepSense architecture proposed by Yao et al. [185].

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, Sensor-based Fusion (**SF**) is applied, a method that considers each sensor individually in order to extract sensor-specific information [50]. 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

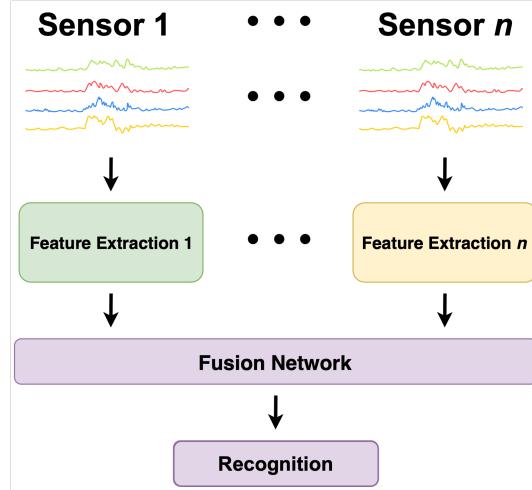


Figure 3.6: The model architecture in a nutshell: Using subnetworks for feature extraction of the different sensors (accelerometer, gyroscope, and GPS) and afterwards a fusion network to combine the features for final incident recognition [35].

third convolutional layers apply zero-padding which differs from the original DeepSense framework proposed by Yao et al. [185]. 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 [163]. The described convolutional layers are complemented by batch normalization layers to reduce internal covariate shifts [78], 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 [69]. Residual blocks have been applied with great success to overcome this issue [69].

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 3.6.

The last big component of CycleSense is a RNN. RNN architectures such as Long Short-Term Memory (LSTM) [72] or Gated Recurrent Units (GRUs) [41] 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

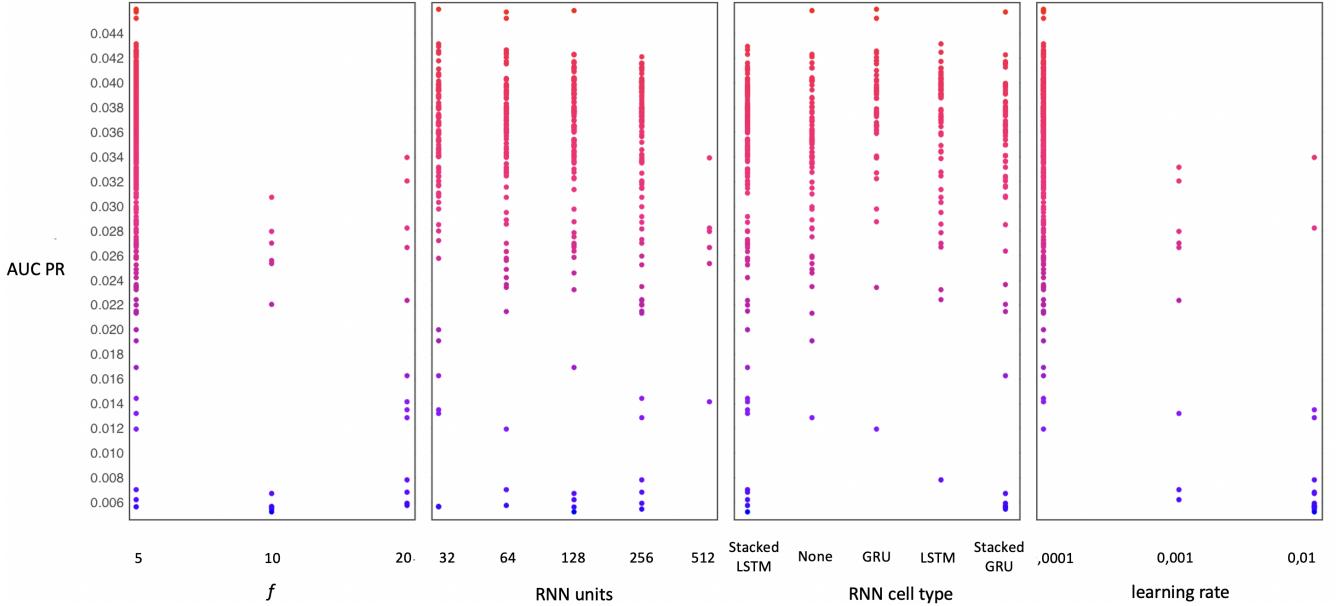


Figure 3.7: Results of the hyperparameter optimization for the four variables f (the window size), the number of RNN units used, the RNN cell type utilized, and the learning rate from left to right.

possible to identify patterns or relationships inside the timestamps of a bucket or between buckets. Similar to Yao et al. [185], we also chose stacked GRU cells as they efficiently improve the model capacity [60].

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 3.7.

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 GitHub⁹.

3.3.3 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

⁹<https://github.com/simra-project/CycleSense>

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 [4.6](#)) 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 [\[182\]](#), 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 [3.5](#).

3.4 Evaluation

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 3.1. We give an overview of the changes we made to the baseline methods with the goal of a fair comparison in Section 3.4.1. We also describe the metrics that we use to compare our model to the baseline methods (Section 3.2.2) before presenting the results of our evaluation (Section 3.4.3).

3.4.1 Baselines

The first baseline is our original heuristic [88] 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 [145]. We used the original preprocessing pipeline (which differs significantly from the here presented one) but used the full data set as introduced in Section 3.2.1. We skipped the under-sampling step, disregarded the phone location and the bike type feature for the reasons mentioned in Section 3.3.1, and used a non-overlapping sliding window approach with 10 second windows for better comparability.

The third baseline is DeepSense [185], which we implemented and trained as the authors describe in their work. For the differences between DeepSense and CycleSense, see sections 3.3.2 and 3.6.

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.

3.4.2 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 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 [53] while also reporting the respective confusion matrices.

3.4.3 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](#) [80]. All of these were trained on the Android data set consisting of more recent rides which provides the best results for all approaches. Figure 3.8 and Table 3.2 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](#) [80]: 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

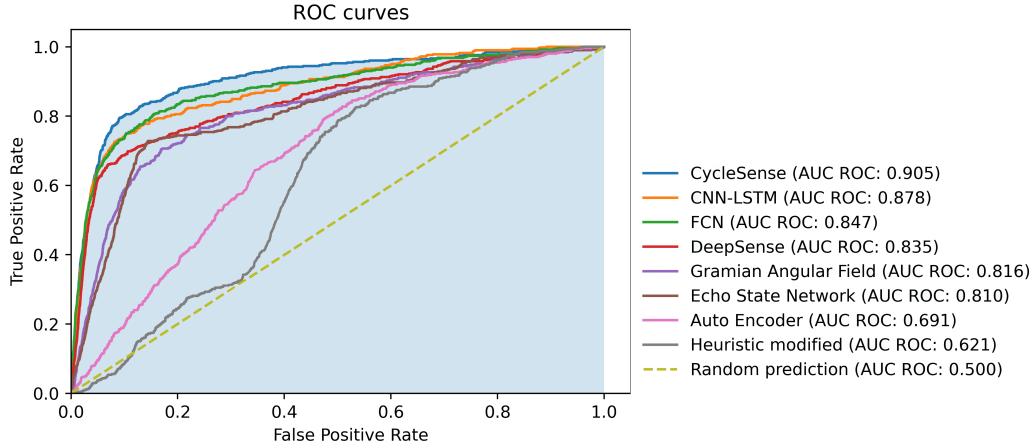


Figure 3.8: Comparison of the baselines and common model architectures used in the context of DL for TSC [80] with the CycleSense model. Note that all of these models have been trained and evaluated on rides contained in the SimRa data set that have been recorded with more recent versions of the SimRa Android app.

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.

So far, we have predominantly focused on newer rides recorded on the Android version of the SimRa app. As shown in Figure 3.9a, 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 3.9b).

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 3.10a. It clearly shows, that CycleSense does not perform as good as in Berlin. Since most regions lack training data, it would not be a

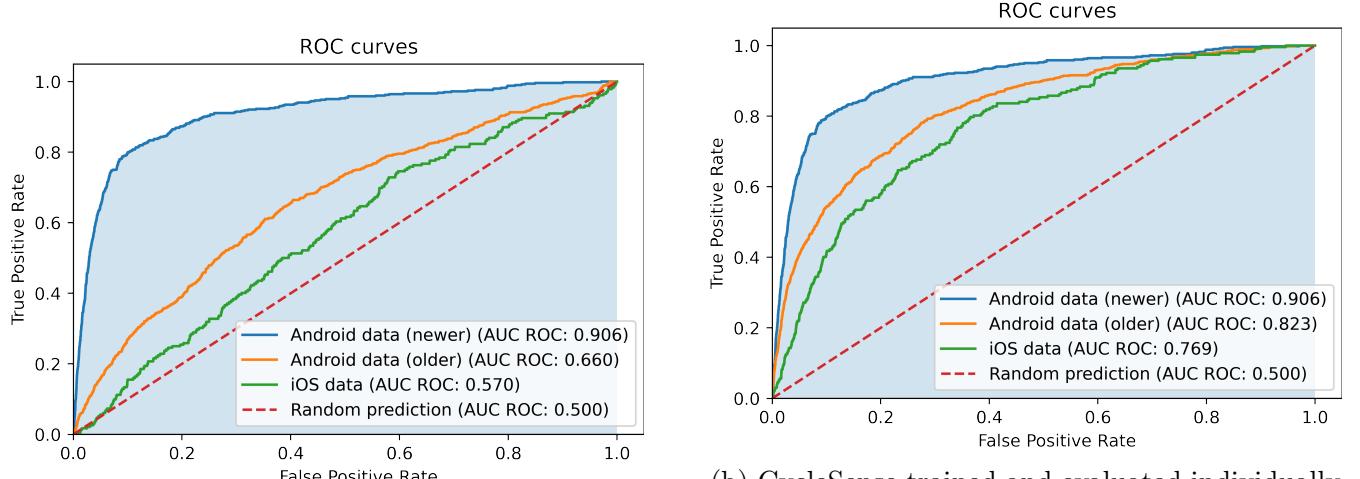
	TN	FP	FN	TP	AUC ROC	Precision	Recall	F1-Score	MCC
CycleSense	107934	10893	104	400	0.906	0.035	0.794	0.068	0.156
CNN-LSTM	106427	12400	127	377	0.878	0.030	0.748	0.057	0.135
FCN	100932	18500	98	402	0.847	0.021	0.804	0.041	0.115
DeepSense	106192	12635	153	351	0.835	0.027	0.696	0.052	0.123
GAF	98782	20045	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 3.2: Comparison of the baselines and common model architectures used in the context of [DL](#) for [TSC](#) [80] with the CycleSense model. See the discussion in Section 3.4.2 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[186] were chosen.

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 3.10b 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.

3.5 Discussion

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. Note that we also discuss some limitations in chapter 7, that may also apply to this approach.



(a) CycleSense trained only on the newer Android rides and evaluated on all parts of the data set.

(b) CycleSense trained and evaluated individually on all parts of the data set.

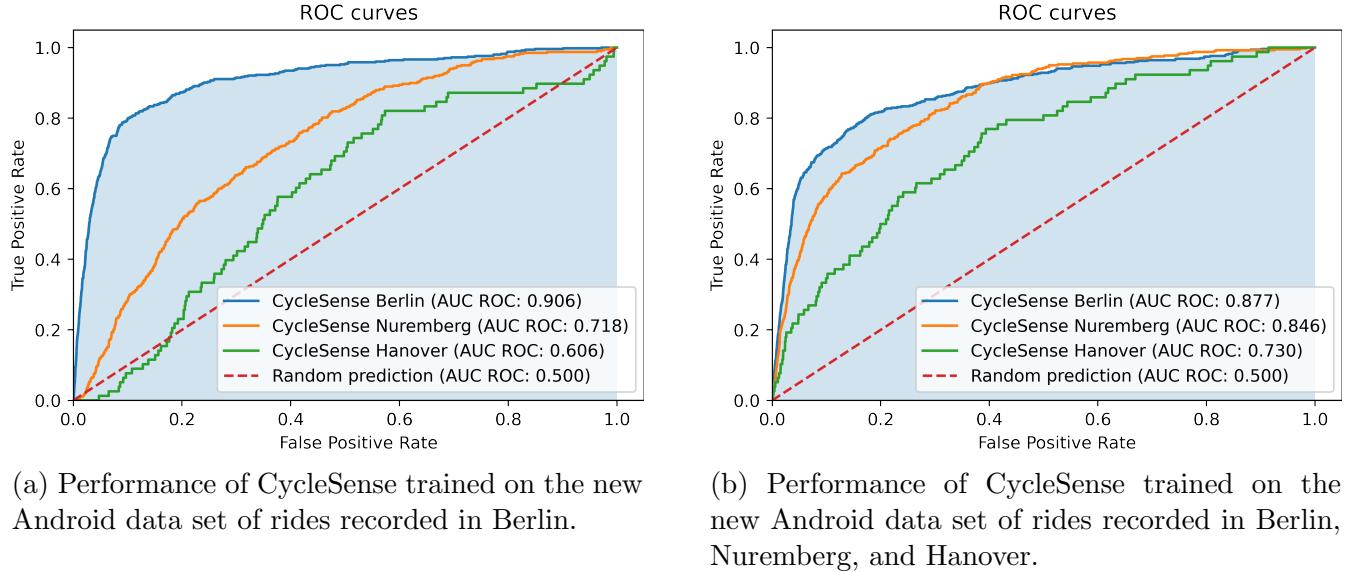
Figure 3.9: Comparison of CycleSense trained only on the newer Android rides or individual CycleSense models trained for each part of the data set. Note that only the newer Android data set was manually cleaned.

3.5.1 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 3.11 illustrates the performance 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.

3.5.2 Technical Limitations of the SimRa Data Set

Recording rates of sensor readings deviate among devices and operating systems and impair model predictions [156], this may also affect the performance of CycleSense: As illustrated in Figure 7.1, 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



(a) Performance of CycleSense trained on the new

Android data set of rides recorded in Berlin.

(b) Performance of CycleSense trained on the new Android data set of rides recorded in Berlin, Nuremberg, and Hanover.

Figure 3.10: Comparison of the performance of CycleSense trained on Berlin and trained on the other regions combined.

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 [88] 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 7.2. Both 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.

3.5.3 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

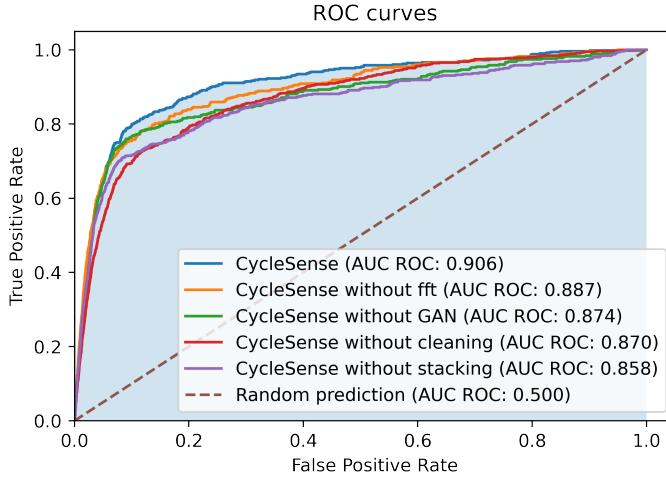


Figure 3.11: Impact of different preprocessing and training steps on the performance of CycleSense.

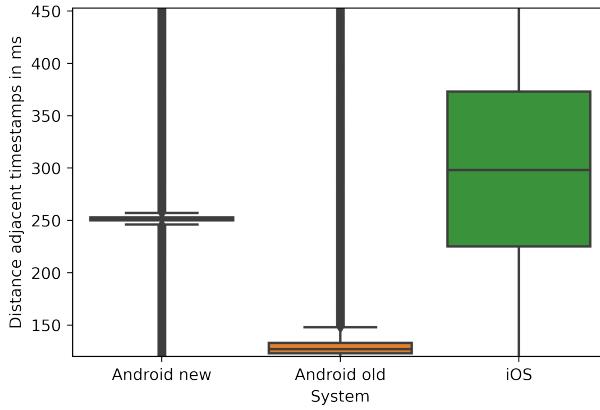


Figure 3.12: Box plot showing the empirical measurement times observed in the SimRa data set for rides recorded with different versions of the SimRa app.

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, 88] 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 sensors. In the SimRa project, this is already happening through an integration of OpenBikeSensor (OBS)¹⁰ data which measures the passing distance of cars. The current data set, however, only includes very few OBS-supported rides.

¹⁰<https://www.openbikesensor.org/>

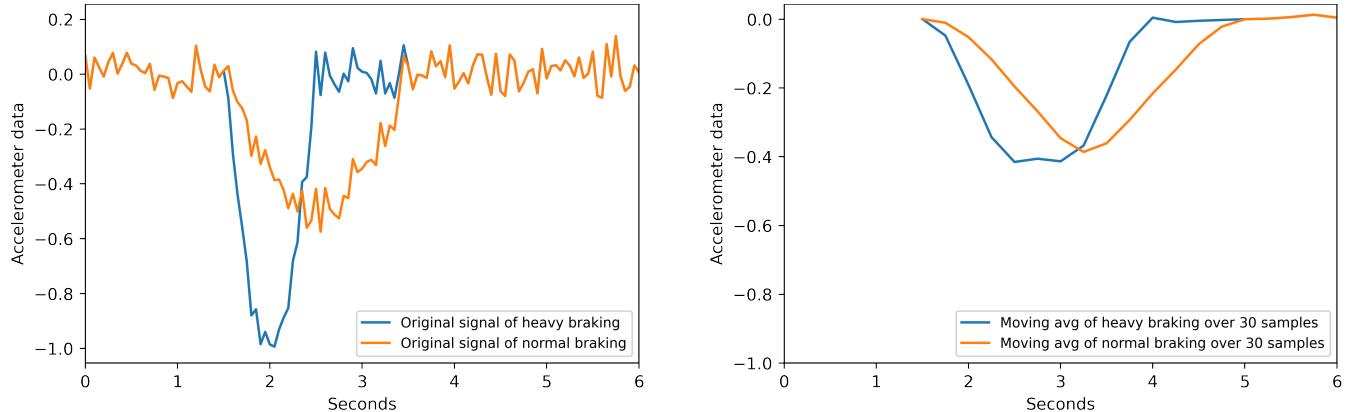


Figure 3.13: Visualization of the sensor data of a simulated heavy braking incident vs. a moderate braking event before and after the moving average has been applied.

In contrast to our classification task, the inherent classification problem of the HAR data set [5], 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.

3.5.4 Data Set Shift

A major challenge in the context of crowdsourced data are data set shifts. Depending on the device (see also subsection 3.5.2), but also on other factors, such as the city of origin of the recorded data (see also subsection 3.4.3), 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 [158], *label shifts*, meaning shifts in the distribution of targets [108], or mixtures thereof. Detecting these shifts [1, 13, 26, 135, 137] and predicting [147] or reducing their impact on the generalization performance is an active field of research [21, 148]. Some approaches aim at model specific improvements to alleviate data set shift [158]. 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 [21]. 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 [43]. In our work we follow this line of thought of a data centric AI approach.

3.6 Alternative Approaches

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 [28], those approaches have constraints such as hand-crafted feature extraction, being able to only learn shallow features [184], or the requirement for large amounts of well-labeled data for model training [175]. Deep Learning techniques and more specifically [CNNs](#) have recently proven to overcome these issues and deliver convincing results in the context of [HAR](#) [143, 175]. Also, implementations of [RNNs](#) such as [LSTM](#) [160, 185] have proven to be successful. Yao et al. [185] use a combination of [RNNs](#) and [CNNs](#) on top of a [SF](#) approach after preprocessing their input data using a [DFT](#).

3.7 Summary

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 [88] 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 chapter, 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.

Chapter 4

Deriving Road Surface Quality from Bicycle Ride Data

In this chapter we present an approach for deriving the surface quality (or rather the lack of), which can be measured as vibrations or bumps, using the **IMU** built in smartphones. 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 crowd-sourcing (e.g., Luedemann et al. [112]), 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.

This chapter contains material published in [89] with 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 crowdsourcing project SimRa [88] (see section 3.2) which previously only focused on incidents.
- 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 [16, 17, 87] 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.

We outlined this chapter as follows: In section 4.1 we give background information about the measurement of road surface quality. We then continue with section 4.2, where we clarify the scenarios and constraints of our approach. We explain our approach and how it is integrated in SimRa (see section 3.2) in section 4.3. After that we describe two ways in which the road surface quality information can be used in section 4.4. We follow this up with an evaluation and discussion of our approach in section 4.5 and section 4.5 respectively. We conclude this chapter in section 4.8 after giving an overview of alternative approaches in section 4.7.

4.1 Road 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.

4.2 Scenarios and Constraints

The idea to use smartphones and crowdsourcing for tracking surface quality is not new. For this reason, we use this dedicated section to briefly discuss the specific goals of our work and the unique constraints of the scenarios we target. We do this to clarify why existing approaches, which we discuss in the next section, cannot solve the problem we target.

First, our work focuses on the experience in terms of “bumpiness” for cyclists. Therefore, we try to quantify the roughness of a road surface and not its type. As an example, a ride on asphalt will usually be smoother than one on a cobblestone road. In practice, though, an asphalt road might have lots of (small) potholes and repair patches while a cobblestone road might use relatively flat (instead of rounded) stones with mostly filled-in gaps between the stones. See fig. 4.1 for an example. In such a setup, a ride on the cobblestone road can be much smoother. Quantifying the



(a) Smooth cobblestones



(b) Rough asphalt

Figure 4.1: Smooth cobblestones can provide a good surface quality while bad asphalt can result in a bumpy ride (Source: Flickr.com).

surface *quality* and not the surface *type* will thus yield different results – a large body of related work is hence not applicable to the problem we target.

Second, due to the width of bicycle tires, a single ride on a street segment will only cover a very small percentage of the surface. Furthermore, cyclists are likely to swerve around the worst potholes and tree roots. Both aspects combined show that it is crucial to base the roughness measurements on a large number of rides, thus, following a “wisdom of the crowds” approach. We hence have to attract a large user group – this has a number of implications.

1. The approach needs to go easy on the phone’s resources (battery, data transmission, compute power, app size) as users will otherwise uninstall or not use the corresponding app. As a result, the approach will have to prefilter data on the phone but is unlikely to run on the phone completely, machine learning-based approaches may be problematic, and the approach cannot expect high resolution raw data.
2. The approach should not have physical setup requirements. Some cyclists mount their phones on the handlebar (which usually will yield the best results for using accelerometer sensors), others keep it in pockets, backpacks, etc. Any approach that *requires* cyclists to use a certain setup will deter a large number of potential users. As an implication, many approaches that work under lab conditions will not work on the street.
3. The approach needs to easily integrate into existing cycling apps. As we have seen in SimRa [88], many cyclists will not use more than one cycling app in parallel. To maxi-

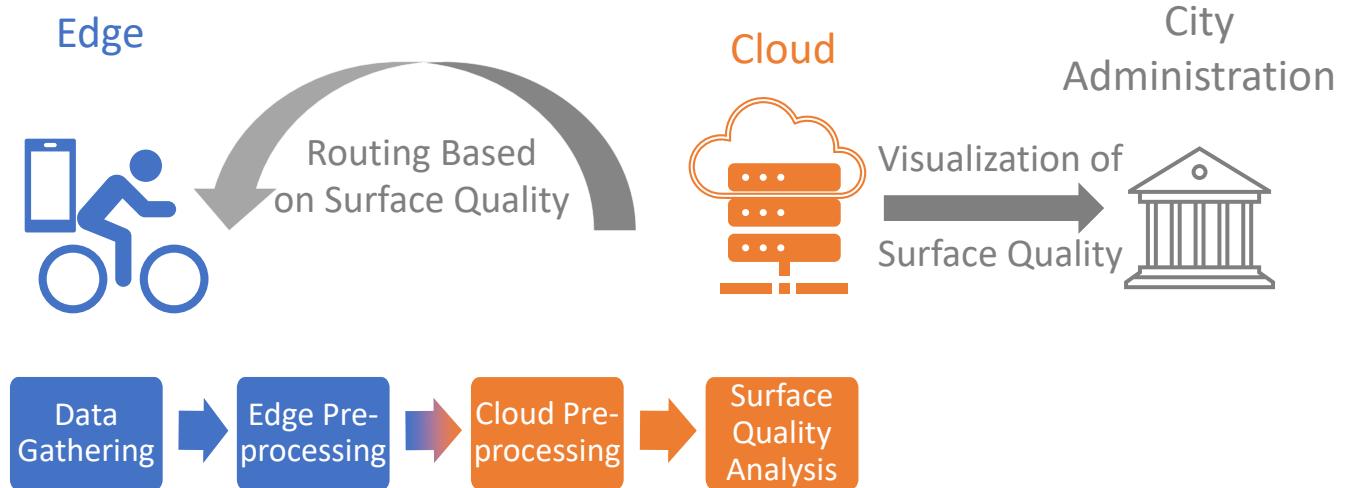


Figure 4.2: Overview of our surface quality analysis pipeline.

mize the potential user base, the approach hence needs to be designed in a way that it can easily be integrated into existing cycling apps. In general, this means that the approach needs to run as an independent background job and may not involve manual labeling and similar activities. They may also be subject to privacy-related restrictions.

The approach we present in this section was designed around these constraints. We retrofitted it to the SimRa app [88] but it could easily be integrated as a plugin in, e.g., Strava or BikeCitizens. As a soft constraint, we tried (and succeeded) to design our approach in a way that it can also process our existing datasets [16, 17, 87].

4.3 Integrating Road Surface Quality Measurement into SimRa

In this section, we start by giving a high-level overview of our data processing pipeline for deriving surface quality (section 4.3.1) before describing how we derive the surface quality (section 4.3.2).

4.3.1 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 4.2. After collecting data in the SimRa app (see section 3.2 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.

4.3.2 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 [88]). 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).

4.4 Using Road Surface Quality Information

In this section, we describe how such surface quality results can be used. In section 4.4.1, 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 section 4.4.2 how we can expose the output data of our pipeline to city administrators.

4.4.1 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. 4.3). We host a modified GraphHopper¹ 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.

4.4.2 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

¹<https://github.com/graphhopper/>

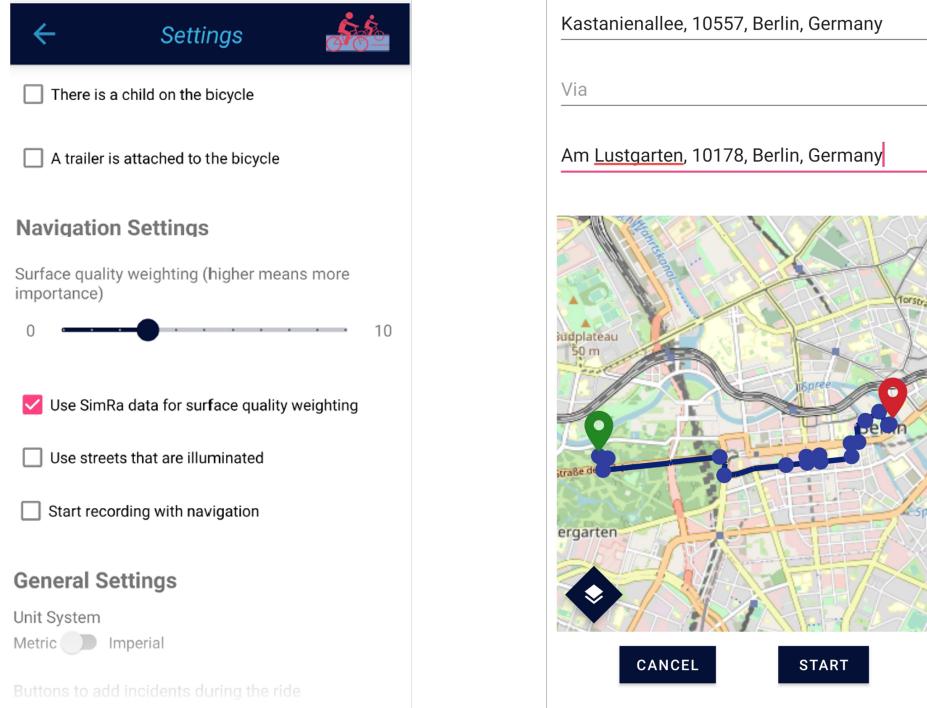


Figure 4.3: With a slider in the settings menu, the weight of the surface quality in the routing can be set.

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 visualization², e.g., with Leaflet³, as depicted in fig. 4.4. 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.

4.5 Evaluation

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 [16, 17, 87] (almost 90,000 rides with more than 650,000km in total⁴). Based on the intuition that different surface

²<https://simra-project.github.io/surfaceQuality/Berlin.html>

³<https://leafletjs.com/>

⁴All numbers in this chapter regarding the SimRa dataset are as of September 2022

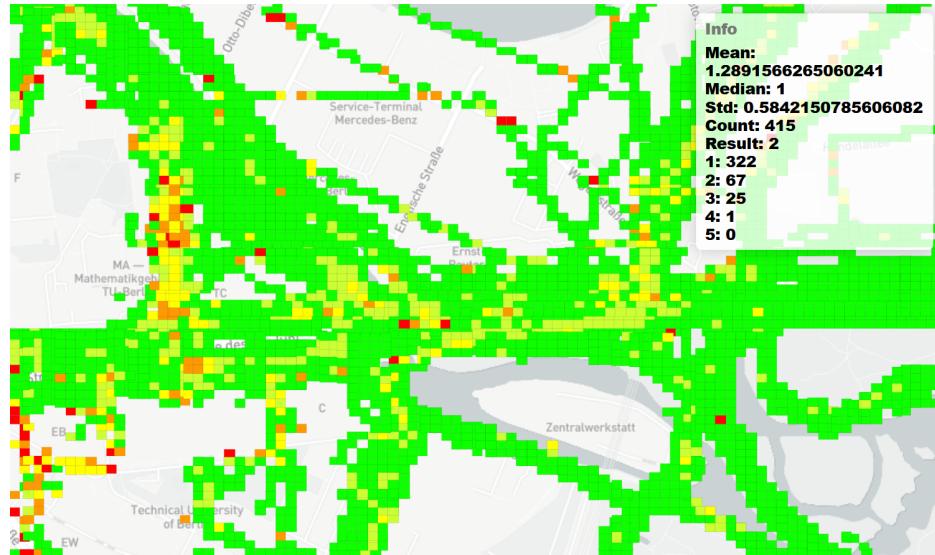


Figure 4.4: A visualization of the output file showing the surface quality of the boxes when hovering over them with the mouse cursor. Color coding is based on the average value.

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 m^2 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 (section 4.5.1). Afterwards, we categorize the additional four “problem spots” as mixed results (section 4.5.2) or seemingly incorrect results (section 4.5.3).

4.5.1 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 [161]: asphalt, flat paving stones, fine gravel, cobblestones. We chose the sections in a way that (i) 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.

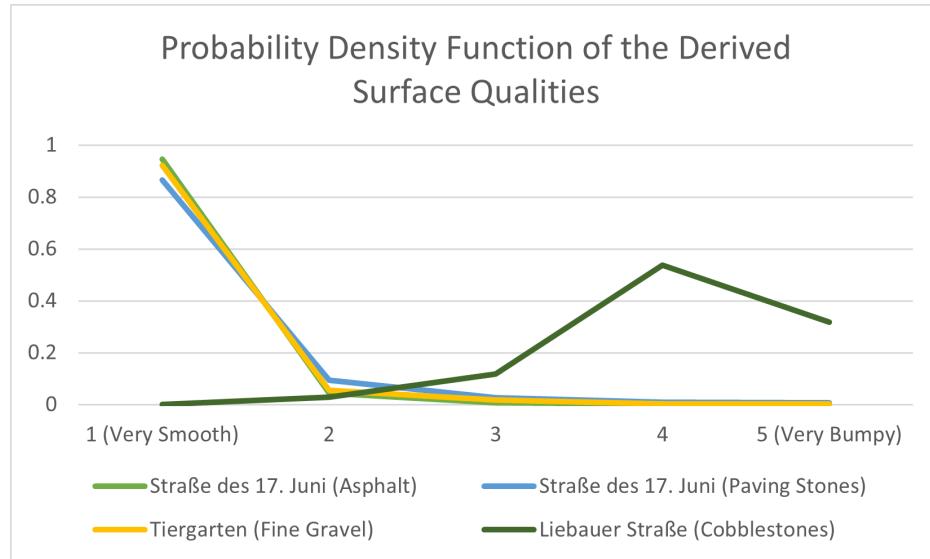


Figure 4.5: The Probability Density Function of the Derived Surface Qualities shows that these segments have very undisputed road surface quality values, since they are either very good or very bad.

table 4.1 and fig. 4.5 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 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 section 4.6). 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.

Table 4.1: 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

Table 4.2: 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

4.5.2 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. 4.6) seem confusing at the first glance.

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. 4.7). 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

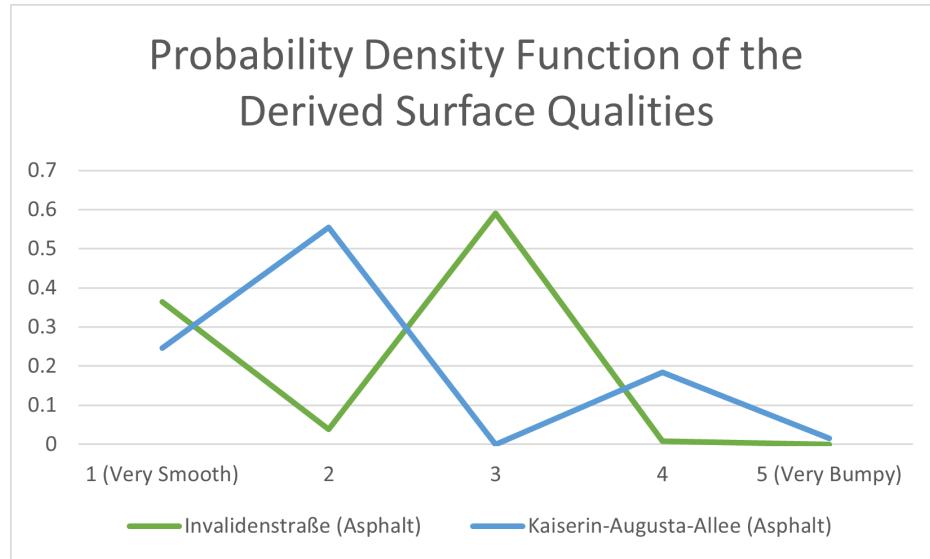


Figure 4.6: The Probability Density Function of the Derived Surface Qualities shows that these segments have confusing road surface quality values: Depending on the ride, they appear to have either very good or very bad surface quality (multiple peaks).



Figure 4.7: A bus lane, a bus station and a bicycle lane on the sidewalk can create different results in a small area. (Source: Apple Maps)

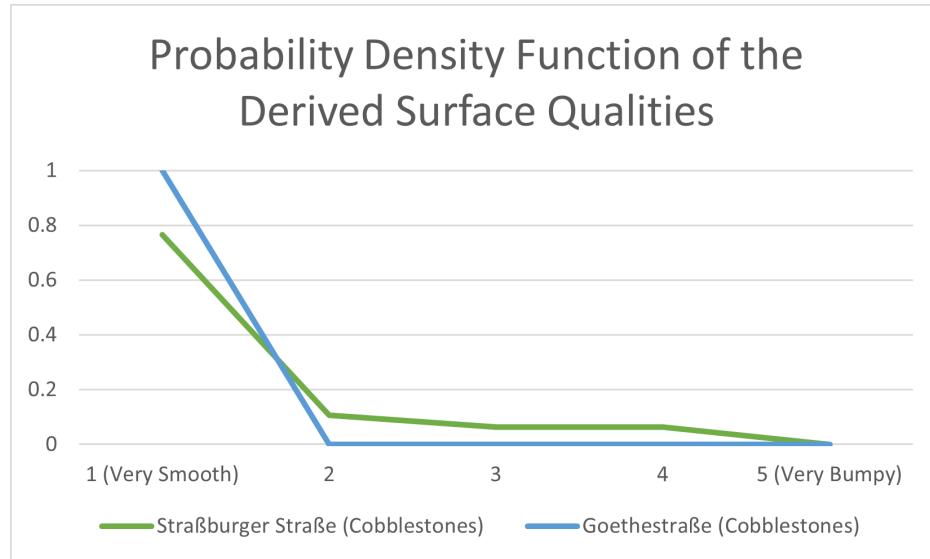


Figure 4.8: The Probability Density Function of the Derived Surface Qualities shows that these segments have very smooth road surfaces, although they are paved with cobblestones.

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 [16, 17, 87] 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 was. Both directions, however, regularly end up in the same 10x10m box as the street is not overly wide, especially considering GPS accuracy.

4.5.3 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 4.3 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. 4.8.



(a) Straßburger Straße



(b) Goethestraße

Figure 4.9: 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)

Table 4.3: 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

The possibility, that this section has a wrong label and is in fact, not paved with cobblestones may come to mind. However, as fig. 4.9 reveals, both streets are indeed paved with cobblestones.

Figure 4.9 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 sidewalk instead. In fact, photos on Apple Maps (not included in this paper) actually show cyclists using the sidewalk instead.

4.6 Discussion

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. Note that we also discuss some limitations in chapter 7, that may also apply to this approach.

4.6.1 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. 4.9 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.

4.6.2 Sensor Inaccuracy

The inaccuracy of the GPS sensors [117] 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.

4.6.3 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.

4.7 Alternative Approaches

Surface quality is usually quantified based on the International Roughness Index (IRI), e.g., [146]. 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 [44] 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. [159] 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. [130] 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. [187] 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. [112] 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. [183] 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., [36], have very specific phone placement requirements, e.g., [45], or focus on detecting the transients, i.e., individual pot holes, e.g., [2, 105, 106].

4.8 Summary

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 crowdsourcing 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 crowdsourcing approach can indeed derive surface quality and implemented two use cases for using such data.

Chapter 5

Creating a Realistic Cyclist Model out of Bicycle Ride Data

In this chapter, we describe how we created a realistic cyclist model out of the SimRa dataset for [SUMO](#). The main idea is, to deduce the lateral movement specifications, i.e. acceleration, deceleration and maximum velocity, as well as the left-turn behavior at intersections of cyclists by analyzing the SimRa dataset and adapting it to [SUMO](#).

In [SUMO](#), vehicles and their dynamics are simulated individually [111]. 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. [93] investigated the intersection behavior of cyclists using camera traces and transferred findings into [SUMO](#). Also, Grigoropoulos et al. [62] improved modeling of bicycle infrastructure at intersections while Heinovski et al. [71] created a virtual cycling environment to import real bicycle behavior directly into [SUMO](#). Nevertheless, the current cyclist behavior in [SUMO](#) is still rather unrealistic; so far, researchers have devoted much more effort to car models, e.g., [9, 33, 57, 59, 96, 103, 144, 164]. 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 SimRa¹ [88] have made thousands of cycle tracks available as open data.

¹<https://github.com/simra-project/>

In this chapter, 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. This chapter combines material published in [90, 91] and contains 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.

This chapter is organized in the following manner: section 5.1 has background information on **SUMO**. section 5.2 contains an analysis and comparison of cyclists’ lateral movement and behavior at intersections in real life (SimRa dataset) and **SUMO**. section 5.3 gives a description of how we implemented the insights gained from the analysis from the previous section. section 5.4 evaluates and section 5.5 discusses our approach. section 5.6 gives an overview of alternative approaches and section 5.7 concludes this chapter.

5.1 **SUMO** - Simulation of Urban Mobility

SUMO is an open source traffic simulation tool that offers macroscopic as well as microscopic simulation of vehicle mobility [111]. **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., [14].

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 [62].

Table 5.1: 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

5.2 Cycling Behavior in SimRa and SUMO

In this section, we analyze real-world cyclists’ behavior extracted from the SimRa dataset² and compare it to the behavior of SUMO’s default bicycle model. SimRa’s dataset stems from crowd-sourced smartphone data generation and thus suffers from poor sensor quality [38, 171] as well as heterogeneous hardware and users [11]. This leads to a lot of unclean data, which we first need to filter out (section 5.2.1). 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 section 5.2.2. We then analyze acceleration, deceleration and velocity behavior for each of the four models in sections 5.2.3 to 5.2.5 before discussing left-turn behavior at four-way intersections of the different models in section 5.2.6. 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 [16, 17, 87] and more recent rides available on GitHub,³ we also used non-public rides which have, for privacy reasons, not been published yet. Table 5.1 summarizes the most important attributes of the dataset that we used.

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

²All numbers in this chapter regarding the SimRa dataset are as of January 2023

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

5.2.1 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.⁴

5.2.2 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 5.1 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.

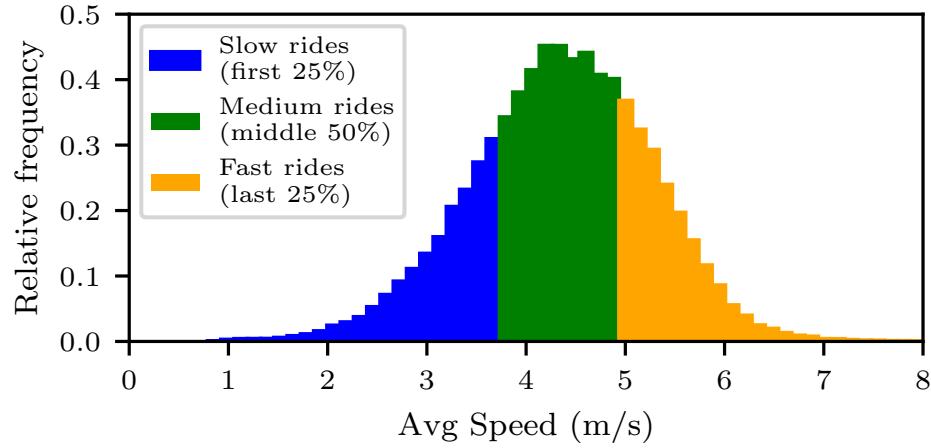


Figure 5.1: Histogram of the empirical average velocity capabilities of cyclists found in the SimRa dataset. The average velocity of all bicycle rides after the preprocessing is 4.38 m/s with a standard deviation of 0.89 and a median of 4.42 m/s.

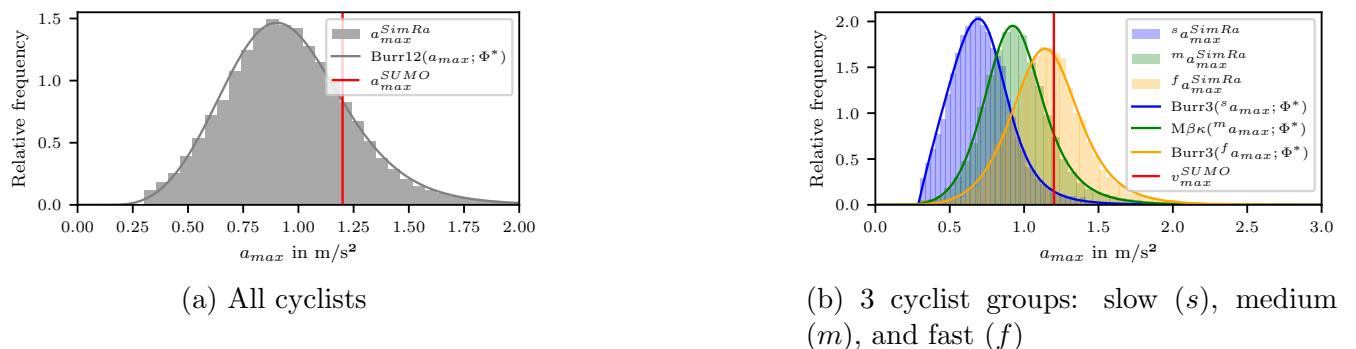


Figure 5.2: Histogram of maximum acceleration capabilities found in the empirical SimRa dataset and their respective distribution functions. The red scalar represents the default value in [SUMO](#).

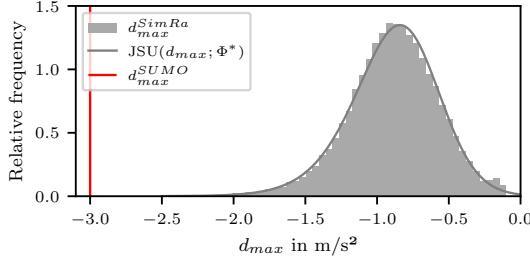
5.2.3 Acceleration

For analyzing cyclist acceleration, we extracted acceleration maneuvers from the dataset. For this, we slightly adapted the approach of [114]. 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 SciPy⁵, showed that the Burr (Type XII) distribution [29] $Burr12(x; c, d) = c * d * \frac{x^{c-1}}{(1+x^c)^{d+1}}$ for $x = a_{max} \geq 0$ and $c, d > 0$ fits the data best for the general cyclist model (see also fig. 5.2a). For the models of the slow and fast cyclist models, the Burr (Type III) distribution [29]

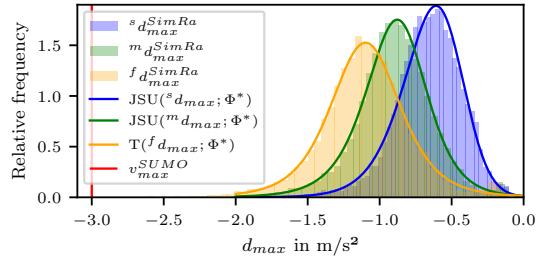
$Burr3(x; c, d) = c * d * \frac{x^{-c-1}}{(1+x^c)^{d+1}}$ for $x = a_{max} \geq 0$ and $c, d > 0$ fits the data best, while the Mielke Beta-Kappa distribution [118]

$M\beta\kappa(x; c, d) = \frac{k*x^{k-1}}{(1+x^s)^{1+k/s}}$ for $x = a_{max} > 0$ and $k, s > 0$ is the best fit for the medium cyclist model's acceleration distribution (see also fig. 5.2b).

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/s². 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/s² 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.



(a) All cyclists



(b) 3 cyclist groups: slow (s), medium (m), and fast (f)

Figure 5.3: 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**.

5.2.4 Deceleration

For analyzing cyclist deceleration, we extracted deceleration maneuvers from the dataset (see section 5.2.3). Distribution fitting processes showed here that the Johnson's S_U -distribution [84] $JSU(x; a, b) = \frac{b}{\sqrt{x^2+1}}\phi(a+b*\log(x+\sqrt{x^2+1}))$ 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. 5.3a). $JSU(d_{max}; a, b)$ 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 [157] $t(x; \nu) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\pi*\nu*\Gamma(\nu/2)}} * (1 + x^2/\nu)^{-(\nu+1)/2}$ for $x = d_{max}$ (see also fig. 5.3b).

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.

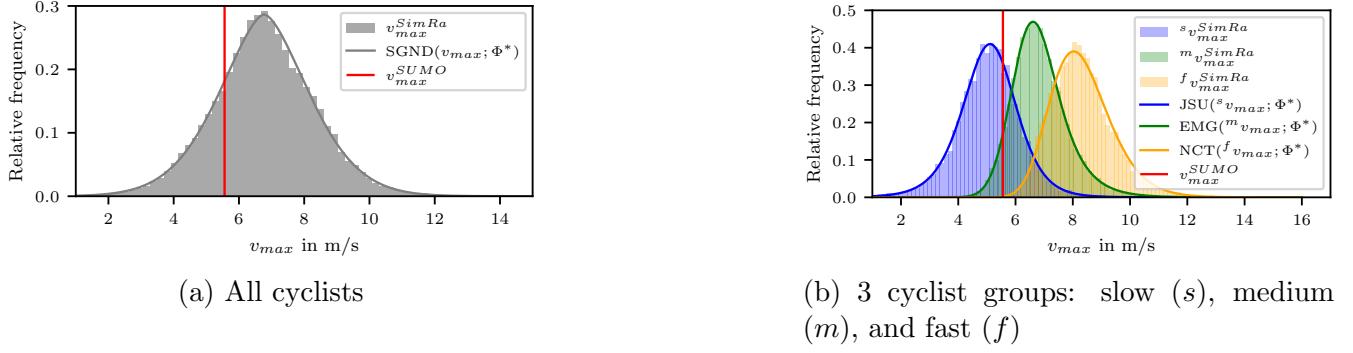


Figure 5.4: Histogram of maximum velocity capabilities found in the empirical SimRa dataset and their respective distribution functions. The red scalar represents the default value in **SUMO**.

5.2.5 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 distribution fitting, we found that the symmetric generalized normal distribution [120] $SGND(x; \beta) = \frac{\beta}{2*\Gamma(1/\beta)} * exp(-|x|^{\beta})$ for $x = v_{max}$ fits the empirical data of all cyclists best (see also fig. 5.4a) 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 [84] $JSU(v_{max}; \Phi^*)$ fits the data best, while the best fit for the medium cyclists' model is the Exponentially Modified Gaussian distribution [64] $EMG(x; K) = \frac{1}{2*K} * exp(\frac{1}{2*K^2} - x/K)erfc(-\frac{x-1/K}{\sqrt{2}})$ for $x = v_{max}$ and $K > 0$. Finally, the non-central t -distribution [73] $NCT(v_{max}; \Phi^*)$ 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.

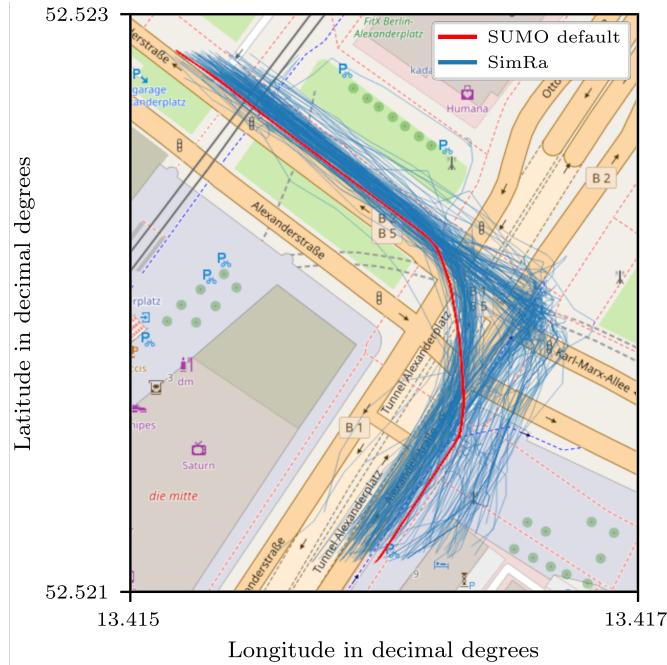


Figure 5.5: Qualitative comparison between the **SUMO** default intersection model and real world data given by SimRa for the intersection between Alexanderstraße and Karl-Marx-Allee in Berlin. SimRa shows two distinct left-turn paths (i.e., a direct and an indirect one) whereas **SUMO** default only models the direct path.

5.2.6 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. 5.5), 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

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

⁵<https://scipy.org/>

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.

5.3 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 section 5.2. 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 section 5.2. One model is for all cyclists combined and the three other models are for modeling slow, medium and fast cyclists.

5.3.1 Longitudinal Kinematic Behavior

In section 5.2, 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 sections 5.2.3 to 5.2.5. 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 [51]. 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.

5.3.2 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 Kaths and Grigoropoulos [93] 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 section 5.2.6 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 section 5.2.6).

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.

5.3.3 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 parameters considered are maximum acceleration (*accel*), maximum deceleration (*decel*) and maximum velocity (*maxSpeed*), splitted into 3 groups, as it can be seen in sections 5.2.3 to 5.2.5. In order to leverage distribution parameters for a bicycle model, we used the *vTypeDistribution* 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 section 5.3.2.

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

5.4 Evaluation

In this section, we evaluate the new cyclist models from section 5.3 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 (section 5.4.1) which we used to create and run the scenarios with. We then continue analyzing acceleration (section 5.4.2), deceleration (section 5.4.3), velocity (section 5.4.4), and left-turn behavior at intersections (section 5.4.5) before evaluating the combination of all model extensions (section 5.4.6). 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](#)

5.4.1 Simulation Setup

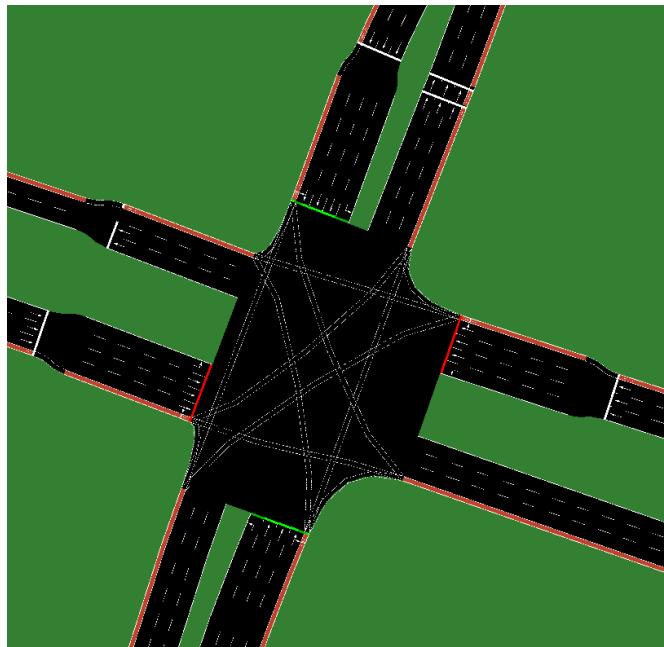
As [SUMO](#) users can import real-world scenarios from OSM data, simulation results can be compared to real-world data and thus be evaluated. For our evaluation, we chose specific traffic scenarios from multiple SimRa regions that are representative and likely to showcase both strengths and weaknesses 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. 5.6), *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 5.2 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, i.e., the maximum acceler-

Table 5.2: Most important attributes of our simulation scenarios.

	Cars	Bicycles	Distance	Lanes
Oranienstr.	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

Figure 5.6: Excerpt from the *Mehringdamm* scenario in [SUMO](#). The scenario was created using OSM data only.

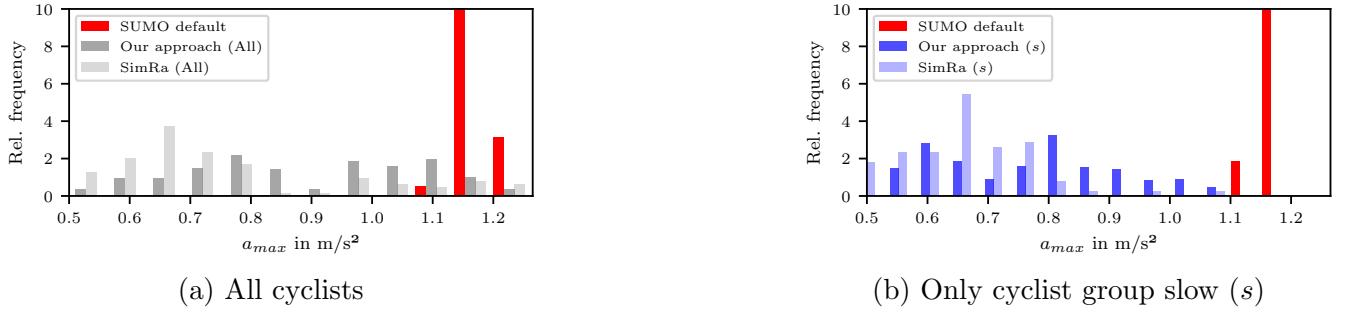


Figure 5.7: Histogram of **SUMO**'s, SimRa's, and our approach's observed maximum accelerations inside the example *Frauentorgraben* scenario. While the maximum accelerations are heterogeneously distributed in the real-world data and our approach, the default values are clustered. Here, the observed accelerations deviate from the configured default value (1.2 m/s^2) due to traffic effects inside the simulation.

ation, 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.

5.4.2 Acceleration

fig. 5.7a shows the empirical distributions of the maximum acceleration among all cyclist types inside the *Frauentorgraben* 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. 5.7b 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. 5.7b 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 5.3. 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

Table 5.3: 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

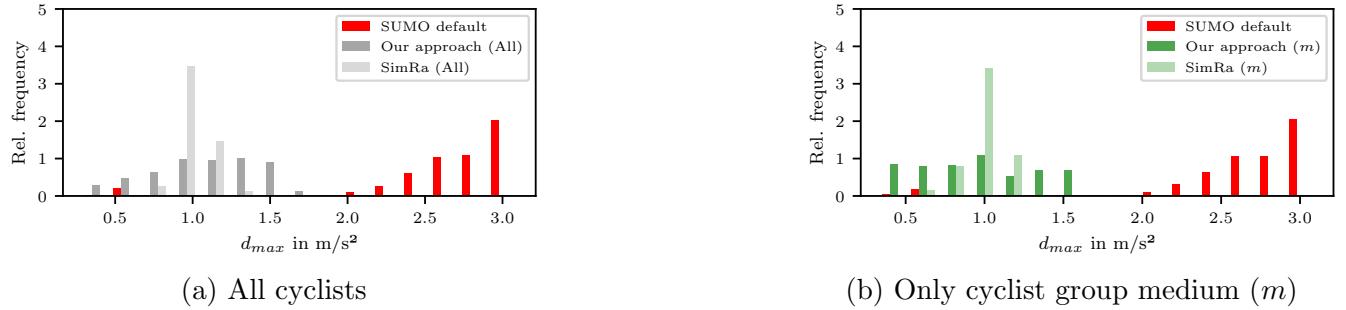


Figure 5.8: Histogram of **SUMO**'s, SimRa's, and our approach's observed maximum decelerations inside the example *Dachauer Straße* scenario. While the maximum decelerations are heterogeneously distributed in the real-world data and our approach, the default values are clustered. Here, the observed decelerations deviate from the configured default value ($3m/s^2$) due to traffic effects inside the simulation.

though, which is probably due to less rides being 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. 5.7b 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.

Table 5.4: 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

5.4.3 Deceleration

fig. 5.8a 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 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. 5.8b 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 5.4 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

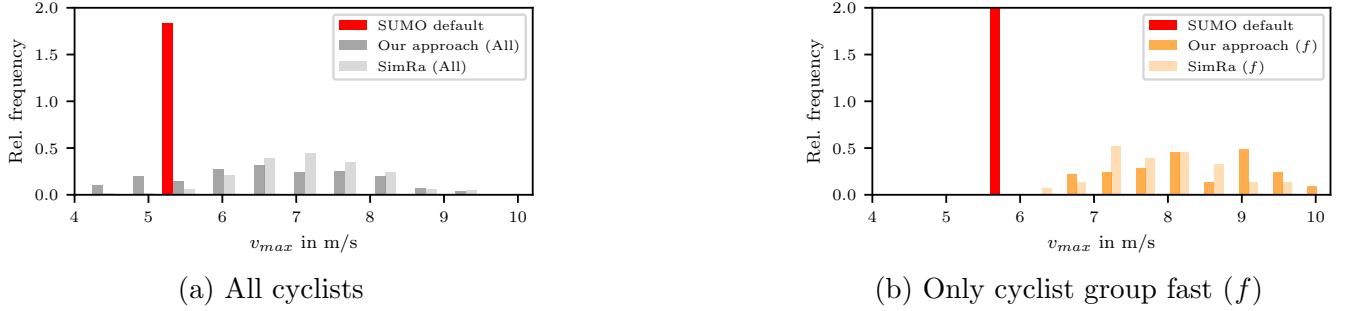


Figure 5.9: Histogram of **SUMO**'s, SimRa's, and our approach's observed maximum velocities inside the example *Oranienstraße* scenario. While the maximum velocities are heterogeneously distributed in the real-world data and our approach, the default values are clustered.

high (above 7 m/s^2), to avoid recording emergency deceleration, which is another parameter in **SUMO**'s vehicle parameterization.

5.4.4 Velocity

fig. 5.9a 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. 5.9b shows exemplary. According to table 5.5 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 data indicates that there are likely to be additional influence factors not captured in our model.

Table 5.5: 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

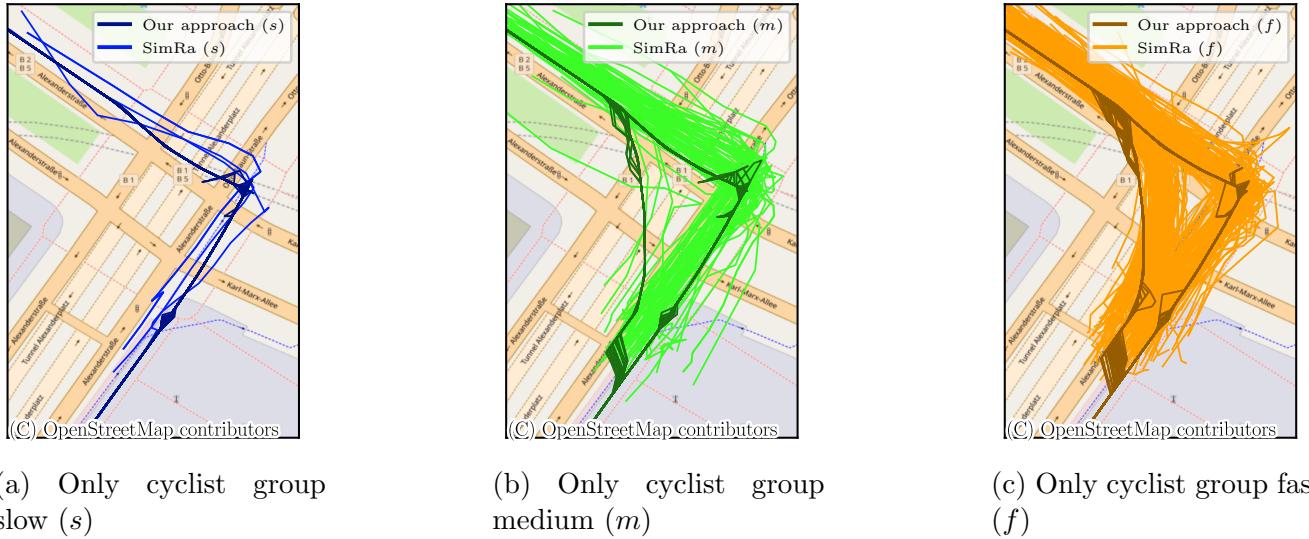


Figure 5.10: Qualitative comparison between the results of our approach for the intersection model and real-world raw GPS data given by SimRa for the intersection between Alexanderstraße and Karl-Marx-Allee in Berlin. SimRa shows two distinct left-turn paths (i.e., a direct and an indirect one), which are also modeled by our models. Also the three different groups have different shares between direct and indirect left-turns.

5.4.5 Left-turn Behavior at Intersections

As shown in fig. 5.10, 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. 5.5, 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.

5.4.6 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. 5.11. 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 section 5.2.6. The “lane only” results in fig. 5.12 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

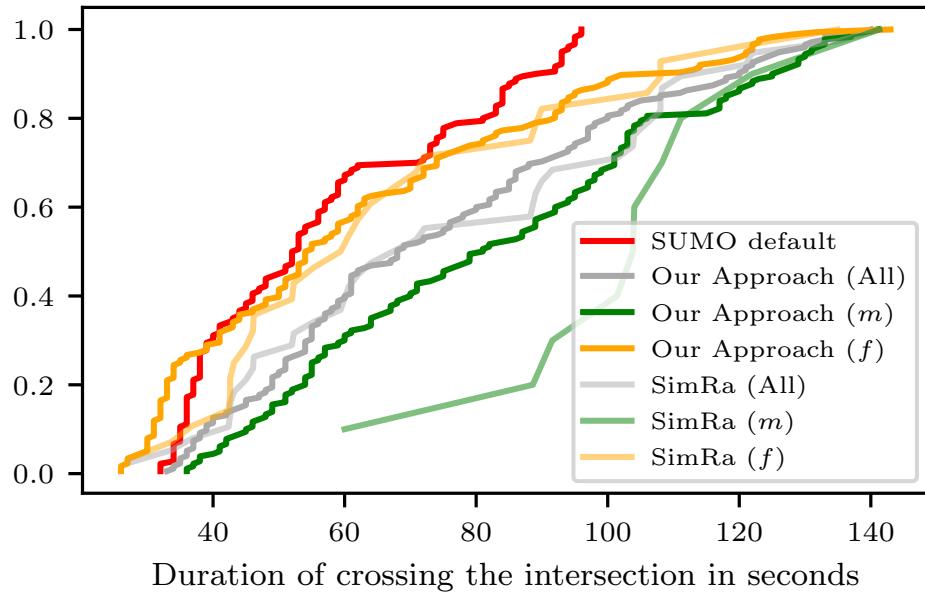


Figure 5.11: ECDFs of the measured durations for crossing the scenario *Warschauer Straße*. It is apparent that our models for all, medium (*m*) and fast (*f*) cyclists outperform **SUMO**'s default as the measured durations converge towards the real-world data. Only our fast cyclist model (*f*) is just marginally better than **SUMO**'s default.

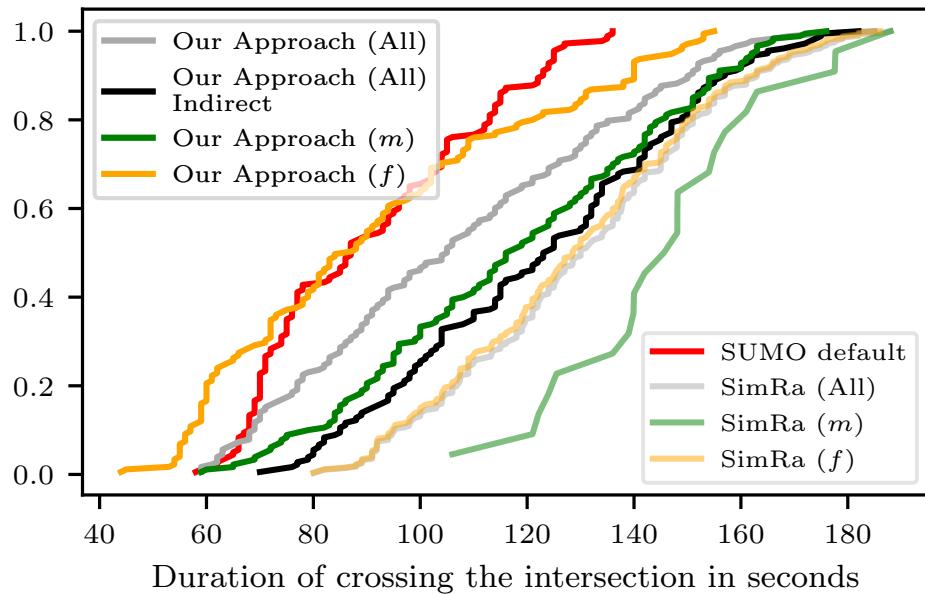


Figure 5.12: ECDFs of the measured durations for crossing the scenario *Mehringdamm*. The results when using our models for all, medium (*m*) and fast (*f*) cyclists are only slightly more realistic than when using the standard **SUMO** model. However, when the *direct* path is blocked for cyclists, the simulation results outperform the default approach.

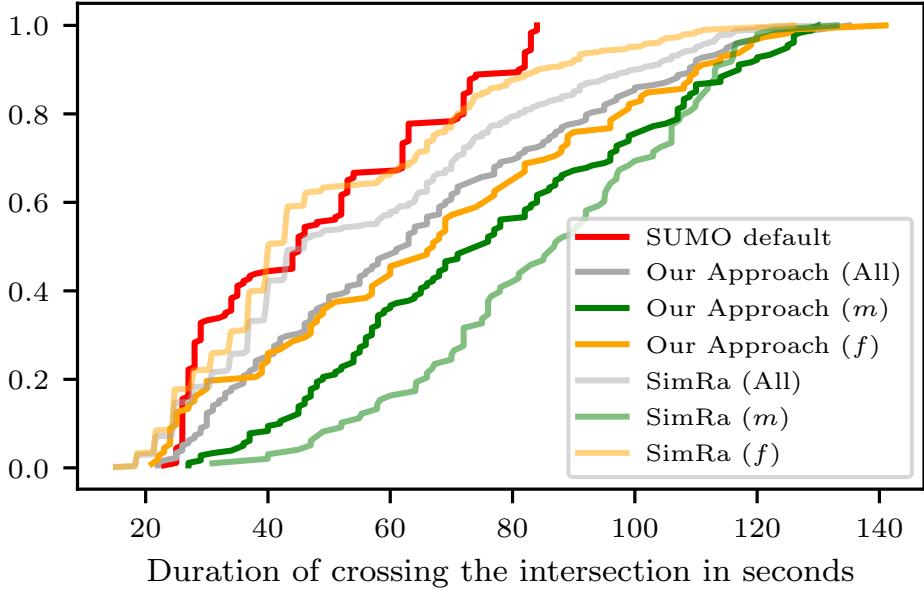


Figure 5.13: ECDFs of the measured durations for crossing the scenario *Alexanderstraße*. Here, our new approach is not more precise than [SUMO](#) default behavior.

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. 5.13). We discuss possible reasons for this in section 5.5.

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. 5.11 to 5.13, since the lines representing the cyclists modeled with our fast cyclist model are the closest to the red line representing [SUMO](#) default.

5.5 Discussion

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 (section 5.5.1) as well as problems resulting from the SimRa dataset as our ground truth data (section 5.5.2). We also describe a potential problem regarding



Figure 5.14: 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.

e-bikes in section 5.5.4. Note that we also discuss some limitations in chapter 7, that may also apply to this approach.

5.5.1 Methodological Challenges

Our initial assumption was that the behavior of cyclists in a single intersection cannot be generalized to all intersections [93] 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. 5.14 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 [93]. These, however, have the inherent limitation that they will either limit the number of bicycles producing data or the number of intersections covered.

5.5.2 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 incidents [88], 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.

5.5.3 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 [92]. This is further aggravated by the fact that the preprocessing step (see section 5.2) 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.

5.5.4 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.

5.6 Alternative Approaches

In this section, we give an overview of related work on improving intersection behavior (section 5.6.1) and longitudinal (section 5.6.2) behavior of cyclists in [SUMO](#)'s simulation models.

5.6.1 Intersection Behavior of Cyclists

Kaths and Grigoropoulos [93] 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 Kaths and Grigoropoulos [93], Grigoropoulos et al. [63] 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. [62] 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.

5.6.2 Longitudinal Behavior of Cyclists

Twaddle and Grigoropoulos [167] 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 [167] 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. [71]. 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.

5.7 Summary

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.

Part III

Conclusions

In this closing part, we conclude our work. We do this by first giving a summary of our three main contributions in chapter 6: (i) The SimRa platform along with the mechanism to automatically derive incidents, (ii) an approach to derive road surface quality from smartphone-based **IMU** data and its integration into SimRa, and (iii) the derivation of a more realistic cyclist simulation for the traffic simulation software **SUMO**. In the last chapter 7, we discuss the general constraints of our work and give an outlook on future work.

Chapter 6

Summary

Increasing the modal share of bicycle traffic comes with a plethora of benefits. The reduction of greenhouse gas emissions has a positive impact both locally and globally. The air, especially in urban areas, gets more clean and this comes with a higher quality of living and lower medical problems. The decrease in greenhouse gas emissions also slow down the global climate crisis, which is one of the main challenges of the 21st century. Another benefit of an increase of bicycle traffic is, is that bicycle infrastructure needs less space than car infrastructure. In many cities, a lack of space leads to a housing crisis and less space for car infrastructure means more space for housing, which in turn can decrease the severity of the housing crisis. These are just some benefits of increasing the modal share of bicycle traffic, which is why both government agencies and citizens want to boost cycling as a means to commute. However, we identified these problems in section 1.1 of chapter 1:

- While there are accident statistics with bicycle accidents, there are no information about incidents. This information is needed, however, to understand where and why incidents accumulate in bicycle traffic. Knowing where and why incidents accumulate, city planners can initiate changes in the infrastructure to make cycling more safe and attractive, since safety concerns are one of the main reasons for people not to cycle or to cycle less than they would want to. Even without any changes to the infrastructure, cyclists can circumvent

the dangerous spots or be more vigilant while passing through them, if they know about them. However, gathering this information the traditional way, with surveys and studies is expensive and difficult to do right, so a possible solution needs to be easy and resource-saving to implement.

- Similar to safety concerns, a lack of comfort is also detrimental to cycling. Cycling comfort is heavily influenced by the quality of the cycling infrastructure, especially the surface quality thereof. So, an efficient means of measuring the road surface quality of cycling infrastructure can help in the same manner as before, by helping city planners detect the most urgent spots to be improved and by informing cyclists, who in turn can adjust their cycling route.
- Having the information about hazardous hotspots in bicycle traffic is not enough though, to act on it. Since changes to traffic infrastructure are expensive to undertake, these changes need to be planned thoroughly and for changes to be approved by the administration, they need to be sure, that the changes will in fact improve the traffic situation as desired. This is where traffic simulation software can be used. The new infrastructure can be modeled and, with a realistic simulation, it can be compared to the current state, to see any improvements. A realistic cyclist simulation model is crucial for that, since the insights won't be representative otherwise. However, creating cyclist simulation models is complex, since it is not very easy to obtain the relevant data for that. This arises the need for an approach using existing data to derive realistic cycling models off of it.

To address these three problems, we presented one solution for each of them in our work:

- We created *SimRa*, a platform for detecting hazardous hotspots in bicycle traffic. SimRa uses a crowdsourcing approach, where users of a smartphone application record their cycling trips and after each trip give detailed information about incidents that happened during the ride. They can then upload their ride, which is a time series containing the GPS trace, as well as IMU data, together with the incidents, if any happened during the ride. In fact, the SimRa project can even be considered as a Citizen Science project. Since remembering every incident and its specifics during a ride is a difficult task for the user, we implemented

a feature to automatically detect incidents using a [DL](#) model called *CycleSense*. *CycleSense* can confidently propose exact locations long the cycling trip where an incident might have happened, which makes the reporting of the incidents easier for the users. This, in turn, helps to record every incident that occurred during a ride without forgetting one and due to a better user experience event contributes to a higher user count and thus more representative data. With the SimRa dataset, city planners have an important information source where they need to focus their attention to, when working to improve the bicycle safety of their area of responsibility. On the other hands, citizens can use the data to increase pressure on the policy makers to act and to improve the situation for cyclists. Additionally, they can inform themselves about the dangerous locations in bicycle traffic and avoid them.

- We developed an approach to derive the road surface quality from cycling trip data, that contains GPS and [IMU](#) data readings. Notably, we designed our approach, so that it can easily be integrated in any preexisting solution or dataset containing cycling trip data, fulfilling the requirements mentioned before. We also have shown how we integrated this approach to *SimRa* and created a visualization of the results. With this information, the transportation department can easily track the state of their bicycle infrastructure and organize the maintenance in a more efficient way. Also, have more information to plan their cycling trips.
- We improved the cyclist simulation model of [SUMO](#). We did this by first examining the lateral movement and intersection behavior of cyclists with the SimRa dataset and comparing it with the cyclist simulation model that was already in [SUMO](#). Our analysis has shown, that [SUMO](#)'s cyclist model is very unrealistic, which is hardly surprising, considering the fact, that [SUMO](#) simulates cyclists either as fast pedestrians or slow cars. We then inferred the three different cyclist types *slow*, *medium* and *fast* by splitting the rides of the SimRa dataset into three parts by their speed. For each subset, we calculated the acceleration, deceleration, maximum velocity distributions, as well as their left-turn behavior at intersections. We implemented our findings as [SUMO](#) plugins, so that users can use them in their simulations. With that, city planners, e.g., can use three different cyclist types, which are all more realistic than the default cyclist model of [SUMO](#). In doing so, they can better plan the way they want to change the bicycle infrastructure.

This thesis consists of three parts. Part I, Foundations, begins with an introduction into the topic, a motivation of this thesis, the problem statements, as well as the structure.

Part II, Improving Safety in Bicycle Traffic, contains all of our main contributions. In chapter 3 we introduced *SimRa*, a platform for gathering cycling trip and incident data, as well as our approach to automatically detect incidents from IMU data from cycling trips, called *CycleSense*. We then presented an approach to derive road surface quality from bicycle trip data in chapter 4. There, we also showed how to integrate our approach in an existing cycling trip recording app by integrating the approach into *SimRa* and how the results can be visualized. In chapter 5, which is the last chapter of part II, we showed how we extract parameters for acceleration, deceleration and velocity of cyclists, as well as their left-turn behavior at intersections. We also showed, how we implement our findings as a plugin for SUMO. Chapters 3 to 5 are structured very similarly. They start with a brief introduction, continue with background information, a description, an evaluation, a discussion of the approach, and end with an overview of alternative approaches and a conclusion.

In part III, Conclusions, we concluded this thesis with a summary in chapter 6 and a discussion of our work and and outlook for future work in section 7.3.

Chapter 7

Discussion and Outlook

In this chapter, we discuss limitations of our approach and give an outlook for future work. While we discuss the limitations of each contribution in their respective chapters in part II, there are some limitations, that they share, which we collect here. This chapter contains partially adapted material published in [88–92].

7.1 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 [12, 32, 86]. 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 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. [156] 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. [97] have shown that orientation sensor data is also affected by this variability.

Implications for Gathering Cycling Trip Data and Detecting Incidents

This is in contrast to the HAR tasks, we compare our task to in chapter 3. The HAR data set [5] 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.

Implications for Deriving Road Surface Quality from Cycling Trip Data

Heterogeneous IMU data poses also a challenge when deriving road surface quality, as we do in chapter 4. This comes from the fact that different smartphone models have different GPS modules and motions 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.

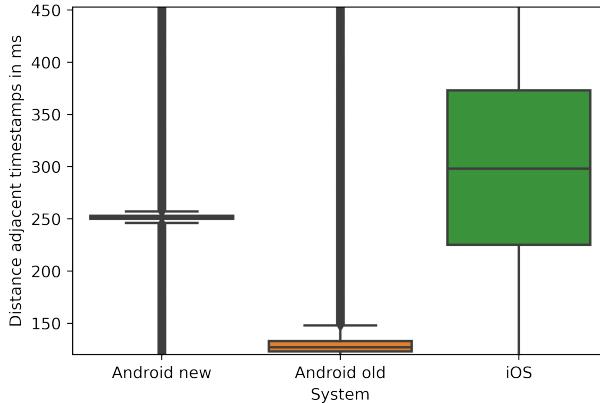


Figure 7.1: Box plot showing the empirical measurement times observed in the SimRa data set for rides recorded with different versions of the SimRa app.

7.2 Limitations of the Preprocessing in SimRa

It is important to note, that SimRa was developed for identifying 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.

Implications for Gathering Cycling Trip Data and Detecting Incidents

Recording rates of sensor readings deviate among devices and operating systems and impair model predictions [156], this may also affect the performance of *CycleSense*: As illustrated in Figure 7.1, 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.

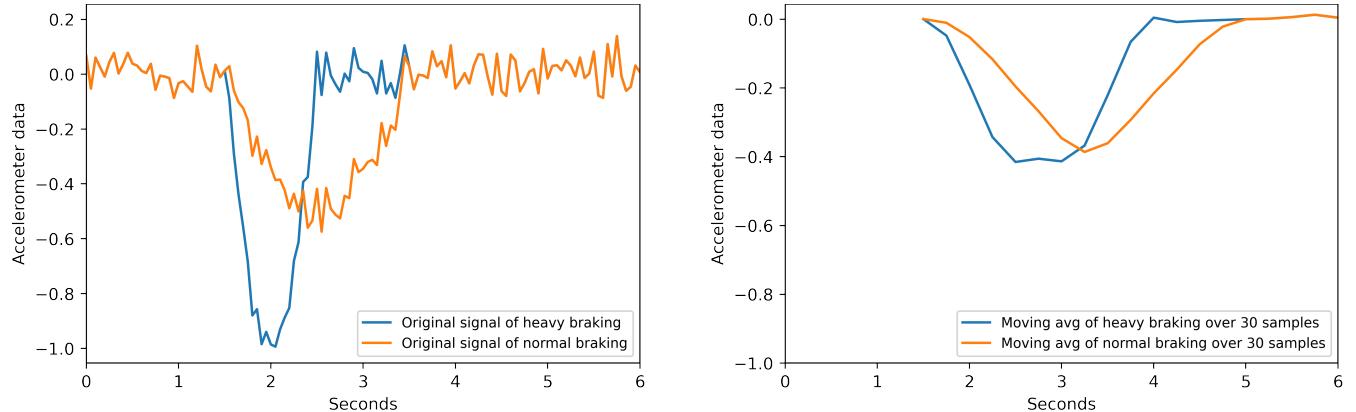


Figure 7.2: Visualization of the sensor data of a simulated heavy braking incident vs. a moderate braking event before and after the moving average has been applied.

Aside from that, the moving average that is used in the SimRa app to condense the data and comply to users' upload volume constraints [88] 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 7.2. Both 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.

Implications for Gathering Cycling Trip Data and Detecting Incidents

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 section 4.5, the surface quality we derive are sufficiently precise for our intended use cases.

7.3 Outlook

A promising direction for future work is to explore the relationship between incidents and traffic infrastructure. Since the *SimRa* dataset contains incidents with their location and incident type,

it might be possible to find out which conditions dangerous spots in traffic have in common. E.g., it seems apparent that a narrow street without a protected bicycle lane would have a high risk for close passes. But such an analysis could reveal far more subtle connections between a certain traffic infrastructure characteristic and an incident type.

Similarly, climate and weather conditions can have an impact on the frequency and type of incidents, which also poses to be a promising topic. Since the *SimRa* dataset contains timestamps of the rides and incidents, the weather information can be considered to see if there are any correlations. E.g., a stormy weather, that impacts the perception of car drivers, could lead them to not see cyclists and thus make a right-turn, cutting the cyclists behind them, that wanted to cycle straight forward.

While we developed and provide visualizations¹ showing the results of the *SimRa* project, further tools can be developed to analyze the dataset in an intuitive manner. In fact, the cities of Walldorf and Wiesloch, as well as the municipalities Zeuthen, Eichwalde and Schulzendorf approached us with a request to develop tools with advanced analysis features for the *SimRa* dataset. This also shows that the *SimRa* project is considered and used by administration bodies to improve the state of their bicycle traffic.

¹<https://simra-project.github.io/dashboard/>

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

IMU Inertial Measurement Unit

IPCC Intergovernmental Panel on Climate Change

IQR Interquartile Range

LSTM Long Short-Term Memory

ML Machine Learning

MLP Multi-Layer Perceptron

NLP Natural Language Processing

OBS OpenBikeSensor

OSM OpenStreetMap

RELU Rectified Linear Unit

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

SUMO Simulation of Urban Mobility

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] Ran An et al. “Weather and cycling in New York: The case of Citibike”. In: *Journal of transport geography* 77 (2019), pp. 97–112.
- [5] 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.
- [6] D. Archer and S. Rahmstorf. *The Climate Crisis: An Introductory Guide to Climate Change*. Cambridge University Press, 2010. ISBN: 9780521407441. URL: <https://books.google.de/books?id=CH5V1Bq9ZnQC>.
- [7] ECSA (European Citizen Science Association). *Ten Principles of Citizen Science*. Berlin, Sept. 2015. DOI: [10.17605/OSF.IO/XPR2N](https://doi.org/10.17605/OSF.IO/XPR2N). URL: [http://doi.org/10.17605/OSF.IO/XPR2N](https://doi.org/10.17605/OSF.IO/XPR2N).
- [8] Witali Aswolinskiy, René Felix Reinhart, and Jochen Steil. “Time series classification in reservoir-and model-space”. In: *Neural Processing Letters* 48.2 (2018), pp. 789–809.

- [9] 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).
- [10] Debrup Banerjee et al. “A deep transfer learning approach for improved post-traumatic stress disorder diagnosis”. In: *Knowledge and Information Systems* 60 (2019), pp. 1693–1724.
- [11] 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.
- [12] Anahid Basiri et al. *Crowdsourced geospatial data quality: Challenges and future directions*. 2019.
- [13] Stephen Bates et al. “Testing for Outliers with Conformal p-values”. In: *arXiv:2104.08279 [math, stat]* (Apr. 2021).
- [14] 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).
- [15] Yoshua Bengio et al. “Generalized denoising auto-encoders as generative models”. In: *Advances in neural information processing systems* 26 (2013).
- [16] 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).
- [17] 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).
- [18] David Bermbach, Erik Wittern, and Stefan Tai. *Cloud service benchmarking*. Springer, 2017.
- [19] Kavi Bhalla et al. “Transport for health: the global burden of disease from motorized road transport”. In: (2014).

- [20] Filippo Maria Bianchi et al. “Reservoir computing approaches for representation and classification of multivariate time series”. In: *IEEE transactions on neural networks and learning systems* 32.5 (2020), pp. 2169–2179.
- [21] Felix Biessmann et al. “Automated data validation in machine learning systems”. In: *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* (2021).
- [22] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [23] Rick Bonney. “Citizen science: A lab tradition”. In: *Living Bird* 15.4 (1996), pp. 7–15.
- [24] Peter R Boyce et al. “Perceptions of safety at night in different lighting conditions”. In: *International Journal of Lighting Research and Technology* 32.2 (2000), pp. 79–91.
- [25] Tamara Bozovic. “Non-walkability in the Car-Centric City”. PhD thesis. Auckland University of Technology, 2021.
- [26] Eric Breck et al. “Data Validation for Machine Learning”. In: *SysML*. 2019, pp. 1–14.
- [27] Eadric Bressel and Brad J Larson. “Bicycle seat designs and their effect on pelvic angle, trunk angle, and comfort”. In: *Medicine & Science in Sports & Exercise* 35.2 (2003), pp. 327–332.
- [28] 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).
- [29] 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).
- [30] Giuseppe Cardone et al. “Crowdsensing in urban areas for city-scale mass gathering management: Geofencing and activity recognition”. In: *IEEE Sensors Journal* 14.12 (2014), pp. 4185–4195.
- [31] José M Cecilia et al. “Mobile crowdsensing approaches to address the COVID-19 pandemic in Spain”. In: *IET Smart Cities* 2.2 (2020), pp. 58–63.
- [32] Abhijnan Chakraborty et al. “Who makes trends? understanding demographic biases in crowdsourced recommendations”. In: *AAAI ICWSM 2017*. Vol. 11. 1. 2017.

- [33] 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/opre.6.2.165](https://doi.org/10.1287/opre.6.2.165).
- [34] Zhengping Che et al. “Boosting deep learning risk prediction with generative adversarial networks for electronic health records”. In: *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2017, pp. 787–792.
- [35] Kaixuan Chen et al. “Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities”. In: *ACM Computing Surveys* 54.4 (2021).
- [36] 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.
- [37] Naima Chouikhi, Boudour Ammar, and Adel M Alimi. “Genesis of basic and multi-layer echo state network recurrent autoencoders for efficient data representations”. In: *arXiv preprint arXiv:1804.08996* (2018).
- [38] 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).
- [39] Ana Karina Christ et al. “Perceiving objective cycling safety: a systematic literature review”. In: *Transportation research procedia* 72 (2023), pp. 1380–1387.
- [40] Henri HCM Christiaans and Angus Bremner. “Comfort on bicycles and the validity of a commercial bicycle fitting system”. In: *Applied ergonomics* 29.3 (1998), pp. 201–211.
- [41] Junyoung Chung et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling”. In: *arXiv preprint arXiv:1412.3555* (2014).
- [42] Vladimir Coric and Marco Gruteser. “Crowdsensing maps of on-street parking spaces”. In: *2013 IEEE International Conference on Distributed Computing in Sensor Systems*. IEEE. 2013, pp. 115–122.

- [43] 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.
- [44] Steven L Cumbaa. *Road Profile Study. Final Report*. Tech. rep. 1986.
- [45] 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.
- [46] Noah S. Diffenbaugh and Elizabeth A. Barnes. “Data-driven predictions of the time remaining until critical global warming thresholds are reached”. In: *Proceedings of the National Academy of Sciences* 120.6 (2023), e2207183120. DOI: [10.1073/pnas.2207183120](https://doi.org/10.1073/pnas.2207183120).
- [47] Jonathan DiGioia et al. “Safety impacts of bicycle infrastructure: A critical review”. In: *Journal of safety research* 61 (2017), pp. 105–119.
- [48] 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).
- [49] Melissa Eitzel et al. “Citizen science terminology matters: Exploring key terms”. In: *Citizen science: Theory and practice* (2017), pp. 1–20.
- [50] Wilfried Elmenreich. “Sensor fusion in time-triggered systems”. PhD thesis. Citeseer, 2002.
- [51] 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).
- [52] and European Environment Agency. *The European environment – State and outlook 2020 – Knowledge for transition to a sustainable Europe*. Publications Office, 2019. DOI: [doi/10.2800/96749](https://doi.org/10.2800/96749).
- [53] Tom Fawcett. “An introduction to ROC analysis”. In: *Pattern recognition letters* 27.8 (2006).

- [54] Rosa Félix, Paulo Cambra, and Filipe Moura. “Build it and give ‘em bikes, and they will come: The effects of cycling infrastructure and bike-sharing system in Lisbon”. In: *Case studies on transport policy* 8.2 (2020), pp. 672–682.
- [55] Mohammad H Forouzanfar et al. “Global, regional, and national comparative risk assessment of 79 behavioural, environmental and occupational, and metabolic risks or clusters of risks, 1990–2015: a systematic analysis for the Global Burden of Disease Study 2015”. In: *The lancet* 388.10053 (2016), pp. 1659–1724.
- [56] Raghu K Ganti, Fan Ye, and Hui Lei. “Mobile crowdsensing: current state and future challenges”. In: *IEEE communications Magazine* 49.11 (2011), pp. 32–39.
- [57] 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).
- [58] Yue Geng and Xinyu Luo. “Cost-sensitive convolutional neural networks for imbalanced time series classification”. In: *Intelligent Data Analysis* 23.2 (2019), pp. 357–370.
- [59] 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).
- [60] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [61] Valentina Grasso et al. “Public crowdsensing of heat waves by social media data”. In: *Advances in Science and Research* 14 (2017), pp. 217–226.
- [62] 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).
- [63] 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).

- [64] 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).
- [65] Mordechai (Muki) Haklay et al. “What Is Citizen Science? The Challenges of Definition”. In: *The Science of Citizen Science*. Ed. by Katrin Vohland et al. Cham: Springer International Publishing, 2021, pp. 13–33. ISBN: 978-3-030-58278-4. DOI: [10.1007/978-3-030-58278-4_2](https://doi.org/10.1007/978-3-030-58278-4_2).
- [66] David Hasenfratz et al. “Participatory air pollution monitoring using smartphones”. In: *Mobile Sensing* 1 (2012), pp. 1–5.
- [67] Nima Hatami, Yann Gavet, and Johan Debayle. “Classification of time-series images using deep convolutional neural networks”. In: *Tenth international conference on machine vision (ICMV 2017)*. Vol. 10696. SPIE. 2018, pp. 242–249.
- [68] Takaki Hayashi and Nakahiro Yoshida. “On covariance estimation of non-synchronously observed diffusion processes”. In: *Bernoulli* 11.2 (2005).
- [69] Kaiming He et al. “Deep residual learning for image recognition”. In: *IEEE CVPR 2016*. 2016.
- [70] Florian Heigl et al. “Toward an international definition of citizen science”. In: *Proceedings of the National Academy of Sciences* 116.17 (2019), pp. 8089–8092. DOI: [10.1073/pnas.1903393116](https://doi.org/10.1073/pnas.1903393116). URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1903393116>.
- [71] 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).
- [72] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997).
- [73] 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).
- [74] Dave Horton. “Fear of cycling”. In: *Cycling and society*. Routledge, 2016, pp. 133–152.

- [75] Jeff Howe et al. “The rise of crowdsourcing”. In: *Wired magazine* 14.6 (2006), pp. 176–183.
- [76] Angela Hull and Craig O’holleran. “Bicycle infrastructure: can good design encourage cycling?” In: *Urban, Planning and Transport Research* 2.1 (2014), pp. 369–406.
- [77] Andrey Ignatov. “Real-time human activity recognition from accelerometer data using convolutional neural networks”. In: *Applied Soft Computing* 62 (2018), pp. 915–922.
- [78] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. PMLR. 2015.
- [79] Alan Irwin. “Citizen science: a study of people, expertise, and sustainable development”. In: (1995).
- [80] Hassan Ismail Fawaz et al. “Deep learning for time series classification: a review”. In: *Data mining and knowledge discovery* 33.4 (2019).
- [81] Hassan Ismail Fawaz et al. “Evaluating surgical skills from kinematic data using convolutional neural networks”. In: *Medical Image Computing and Computer Assisted Intervention-MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part IV* 11. Springer. 2018, pp. 214–221.
- [82] Herbert Jaeger. “The “echo state” approach to analysing and training recurrent neural networks-with an erratum note”. In: *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148.34 (2001), p. 13.
- [83] Matthew D Jarvis, Toby Harris, and Laurissa Tokarchuk. “Ubicomp’13 sencity workshop: sensing festivals as cities”. In: *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. 2013, pp. 1331–1334.
- [84] 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).
- [85] Christian Juhra et al. “Bicycle accidents—Do we only see the tip of the iceberg?: A prospective multi-centre study in a large German city combining medical and police data”. In: *Injury* 43.12 (2012), pp. 2026–2034.

- [86] 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).
- [87] 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).
- [88] Ahmet-Serdar Karakaya, Jonathan Hasenburg, and David Bermbach. “SimRa: Using crowd-sourcing to identify near miss hotspots in bicycle traffic”. In: *PMC* 67 (2020).
- [89] Ahmet-Serdar Karakaya et al. “A Crowdsensing Approach for Deriving Surface Quality of Cycling Infrastructure”. In: *2023 IEEE International Conference on Cloud Engineering (IC2E)*. 2023.
- [90] 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).
- [91] Ahmet-Serdar Karakaya et al. “Achieving realistic cyclist behavior in SUMO using the SimRa dataset”. In: *Computer Communications* 205 (2023), pp. 97–107. ISSN: 0140-3664. DOI: <https://doi.org/10.1016/j.comcom.2023.04.015>. URL: <https://www.sciencedirect.com/science/article/pii/S0140366423001342>.
- [92] Ahmet-Serdar Karakaya et al. “CycleSense: Detecting Near Miss Incidents in Bicycle Traffic from Mobile Motion Sensors”. In: *Elsevier Pervasive and Mobile Computing* (2023).
- [93] Heather Kaths 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.
- [94] Haneen Khreis et al. “The health impacts of traffic-related exposures in urban areas: Understanding real effects, underlying driving forces and co-producing future directions”. In: *Journal of Transport and Health* 3.3 (2016), pp. 249–267. ISSN: 2214-1405. DOI: <https://doi.org/10.1016/j.jth.2016.07.002>.
- [95] Xiangjie Kong et al. “Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges”. In: *IEEE Internet of Things Journal* 6.5 (2019), pp. 8095–8113.

- [96] Stefan Krauß. “Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics”. PhD Thesis. Köln, Germany: Mathematical Institute, Apr. 1998.
- [97] 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).
- [98] Martin Längkvist, Lars Karlsson, and Amy Loutfi. “A review of unsupervised feature learning and deep learning for time-series modeling”. In: *Pattern recognition letters* 42 (2014), pp. 11–24.
- [99] Anneka R Lawson et al. “Perception of safety of cyclists in Dublin City”. In: *Accident Analysis & Prevention* 50 (2013), pp. 499–511.
- [100] Chanam Lee and Anne Vernez Moudon. “Neighbourhood design and physical activity”. In: *Building research & information* 36.5 (2008), pp. 395–411.
- [101] Hoesung Lee et al. “CLIMATE CHANGE 2023 Synthesis Report Summary for Policymakers”. In: *CLIMATE CHANGE 2023 Synthesis Report: Summary for Policymakers* (2023).
- [102] Hoesung Lee et al. *Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Tech. rep. Mar. 2023.
- [103] 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.
- [104] Meng Li, Liehuang Zhu, and Xiaodong Lin. “Privacy-preserving traffic monitoring with false report filtering via fog-assisted vehicular crowdsensing”. In: *IEEE Transactions on Services Computing* 14.6 (2019), pp. 1902–1913.
- [105] 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.

- [106] 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.
- [107] Sangdi Lin and George C Runger. “GCRNN: Group-constrained convolutional recurrent neural network”. In: *IEEE transactions on neural networks and learning systems* 29.10 (2017), pp. 4709–4718.
- [108] 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.
- [109] Chien-Liang Liu, Wen-Hoar Hsaio, and Yao-Chung Tu. “Time series classification with multivariate convolutional neural network”. In: *IEEE Transactions on industrial electronics* 66.6 (2018), pp. 4788–4797.
- [110] Liang Liu et al. “Third-eye: A mobilephone-enabled crowdsensing system for air quality monitoring”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.1 (2018), pp. 1–26.
- [111] 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).
- [112] Kai Luedemann and Mario A. Nascimento. “BikeVibes: An App for Crowdsourcing 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).
- [113] Qianli Ma et al. “Functional echo state network for time series classification”. In: *Information Sciences* 373 (2016), pp. 1–20.

- [114] 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).
- [115] Rebecca Fallon Mayers and Troy D Glover. “Whose lane is it anyway? The experience of cycling in a mid-sized city”. In: *Leisure Sciences* 42.5-6 (2020), pp. 515–532.
- [116] Qipei Mei, Mustafa Güл, and Nima Shirzad-Ghaderoudkhani. “Towards smart cities: crowdsensing-based monitoring of transportation infrastructure using in-traffic vehicles”. In: *Journal of Civil Structural Health Monitoring* 10.4 (2020), pp. 653–665.
- [117] 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).
- [118] 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).
- [119] E Minkman, PJ Van Overloop, and MCA Van der Sanden. “Citizen science in water quality monitoring: mobile crowd sensing for water management in The Netherlands”. In: *World Environmental and Water Resources Congress 2015*. 2015, pp. 1399–1408.
- [120] 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).
- [121] Ingar Næss et al. “The number of patients hospitalized with bicycle injuries is increasing—a cry for better road safety”. In: *Accident Analysis & Prevention* 148 (2020), p. 105836.
- [122] Mohsen Nazemi et al. “Studying bicyclists’ perceived level of safety using a bicycle simulator combined with immersive virtual reality”. In: *Accident Analysis & Prevention* 151 (2021), p. 105943.
- [123] Andrew Ng and Michael Jordan. “On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes”. In: *Advances in neural information processing systems* 14 (2001).

- [124] Alphonse Nkurunziza et al. “Examining the potential for modal change: Motivators and barriers for bicycle commuting in Dar-es-Salaam”. In: *Transport policy* 24 (2012), pp. 249–259.
- [125] Robert B Noland. “Perceived risk and modal choice: risk compensation in transportation systems”. In: *Accident Analysis & Prevention* 27.4 (1995), pp. 503–521.
- [126] Henry Friday Nweke et al. “Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges”. In: *Expert Systems with Applications* 105 (2018), pp. 233–261.
- [127] Javier Ortiz Laguna, Angel Garcia Olaya, and Daniel Borrajo. “A dynamic sliding window approach for activity recognition”. In: *UMAP 2011*. Springer. 2011.
- [128] “Paris Agreement”. In: *United Nations Treaty Collection, Chapter XXVII 7. d ()*. URL: https://treaties.un.org/pages/ViewDetails.aspx?src=TREATY&mtdsg_no=XXVII-7-d&chapter=27&clang=_en (visited on 03/28/2019).
- [129] Eleonora Patacchini et al. “Urban sprawl in Europe”. In: *Brookings-Wharton papers on urban affairs* (2009), pp. 125–149.
- [130] 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.
- [131] Jan Hendrik van Petegem, Paul Schepers, and Gert Jan Wijlhuizen. “The safety of physically separated cycle tracks compared to marked cycle lanes and mixed traffic conditions in Amsterdam”. In: *European Journal of Transport and Infrastructure Research* 21.3 (2021), pp. 19–37.
- [132] Thanh-Trung Phan, Skanda Muralidhar, and Daniel Gatica-Perez. “Drinks & crowds: Characterizing alcohol consumption through crowdsensing and social media”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.2 (2019), pp. 1–30.

- [133] Jurairat Phuttharak and Seng W Loke. “A review of mobile crowdsourcing architectures and challenges: Toward crowd-empowered internet-of-things”. In: *Ieee access* 7 (2018), pp. 304–324.
- [134] Stefan Pietschmann et al. “Croco: Ontology-based, cross-application context management”. In: *2008 Third International Workshop on Semantic Media Adaptation and Personalization*. IEEE. 2008, pp. 88–93.
- [135] Neoklis Polyzotis et al. “Data lifecycle challenges in production machine learning: A survey”. In: *SIGMOD Rec.* Vol. 47. 2018, pp. 17–28.
- [136] James Pritchard. “MaaS to pull us out of a car-centric orbit: Principles for sustainable Mobility-as-a-Service in the context of unsustainable car dependency”. In: *Case Studies on Transport Policy* (2022).
- [137] Stephan Rabanser, Stephan Günemann, and Zachary C. Lipton. *Failing loudly: An empirical study of methods for detecting dataset shift*. 2018.
- [138] Md Abdur Rahman and M Shamim Hossain. “A location-based mobile crowdsensing framework supporting a massive ad hoc social network environment”. In: *IEEE Communications Magazine* 55.3 (2017), pp. 76–85.
- [139] Deepta Rajan and Jayaraman J Thiagarajan. “A generative modeling approach to limited channel ECG classification”. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2018, pp. 2571–2574.
- [140] Nikolaos Rapousis and Maria Papadopouli. “Performance analysis of a user-centric crowd-sensing water quality assessment system”. In: *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. IEEE. 2016, pp. 13–18.
- [141] Arpita Ray et al. “A survey of mobile crowdsensing and crowdsourcing strategies for smart mobile device users”. In: *CCF Transactions on Pervasive Computing and Interaction* 5.1 (2023), pp. 98–123.
- [142] Mikko Rinne, Seppo Törmä, and D Kratinov. “Mobile crowdsensing of parking space using geofencing and activity recognition”. In: *10th ITS European Congress, Helsinki, Finland*. 2014, pp. 16–19.

- [143] 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.
- [144] 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).
- [145] Albert Sánchez Fuster. “Detecting Near Miss Incidents in Bicycle Traffic Using Acceleration Sensor Data”. MA thesis. Universitat Politècnica de Catalunya, 2020.
- [146] Michael W. Sayers. “The International Road Roughness Experiment (IRRE) : establishing correlation and a calibration standard for measurements”. In: 1986.
- [147] 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.
- [148] Sebastian Schelter et al. “On Challenges in Machine Learning Model Management”. In: *Bull. IEEE Comput. Soc. Tech. Comm. Data Eng.* (2018), pp. 5–13.
- [149] JP Schepers et al. “Road factors and bicycle–motor vehicle crashes at unsignalized priority intersections”. In: *Accident Analysis & Prevention* 43.3 (2011), pp. 853–861.
- [150] Joan Serra, Santiago Pascual, and Alexandros Karatzoglou. “Towards a Universal Neural Network Encoder for Time Series.” In: *CCIA*. 2018, pp. 120–129.
- [151] Lanyu Shang et al. “CrowdWaterSens: An uncertainty-aware crowdsensing approach to groundwater contamination estimation”. In: *Pervasive and Mobile Computing* 92 (2023), p. 101788.
- [152] Murad M Shoman et al. “Evaluation of cycling safety and comfort in bad weather and surface conditions using an instrumented bicycle”. In: *IEEE access* 11 (2023), pp. 15096–15108.
- [153] Jonathan Silvertown. “A new dawn for citizen science”. In: *Trends in ecology & evolution* 24.9 (2009), pp. 467–471.

- [154] Vijay Sivaraman et al. “HazeWatch: A participatory sensor system for monitoring air pollution in Sydney”. In: *38th Annual IEEE Conference on Local Computer Networks-Workshops*. IEEE. 2013, pp. 56–64.
- [155] Wei Song et al. “Representation learning with deconvolution for multivariate time series classification and visualization”. In: *Data Science: 6th International Conference of Pioneering Computer Scientists, Engineers and Educators, ICPCSEE 2020, Taiyuan, China, September 18-21, 2020, Proceedings, Part I* 6. Springer. 2020, pp. 310–326.
- [156] Allan Stisen et al. “Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition”. In: *ACM SenSys 2015*. 2015.
- [157] 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).
- [158] M Sugiyama and Motoaki Kawanabe Motoaki. *Machine Learning in Non-Stationary Environments Introduction to Covariate Shift Adaptation*. MIT Press, 2012.
- [159] 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.
- [160] 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).
- [161] 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).
- [162] Silvia Tobias et al. “Soil sealing and unsealing: State of the art and examples”. In: *Land degradation & development* 29.6 (2018), pp. 2015–2024.
- [163] Du Tran et al. “Learning spatiotemporal features with 3d convolutional networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2015.

- [164] 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).
- [165] RK Tripathy and U Rajendra Acharya. “Use of features from RR-time series and EEG signals for automated classification of sleep stages in deep neural network framework”. In: *Biocybernetics and Biomedical Engineering* 38.4 (2018), pp. 890–902.
- [166] John W Tukey et al. *Exploratory data analysis*. Vol. 2. Reading, Mass., 1977.
- [167] 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).
- [168] Munenori Uemura et al. “Feasibility of an AI-based measure of the hand motions of expert and novice surgeons”. In: *Computational and mathematical methods in medicine* 2018 (2018).
- [169] United States Congress. *Crowdsourcing and Citizen Science Act of 2016*.
<https://uscode.house.gov/view.xhtml?req=granuleid:USC-prelim-title15-section3724&num=0&edition=prelim>. 2016.
- [170] Graham Upton and Ian Cook. *Understanding statistics*. Oxford University Press, 1996.
- [171] 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).
- [172] Vladimir Naumovich Vapnik, Vlaminir Vapnik, et al. “Statistical learning theory”. In: (1998).
- [173] Félix Jesús Villanueva et al. “Crowdsensing smart city parking monitoring”. In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE. 2015, pp. 751–756.
- [174] Lina Wahlgren and Peter Schantz. “Exploring bikeability in a metropolitan setting: stimulating and hindering factors in commuting route environments”. In: *BMC public health* 12 (2012), pp. 1–16.

- [175] Jindong Wang et al. “Deep learning for sensor-based activity recognition: A survey”. In: *Pattern Recognition Letters* 119 (2019).
- [176] Xiaojie Wang et al. “A city-wide real-time traffic management system: Enabling crowd-sensing in social Internet of vehicles”. In: *IEEE Communications Magazine* 56.9 (2018), pp. 19–25.
- [177] Zhiguang Wang and Tim Oates. “Encoding time series as images for visual inspection and classification using tiled convolutional neural networks”. In: *Workshops at the twenty-ninth AAAI conference on artificial intelligence*. 2015.
- [178] Zhiguang Wang, Weizhong Yan, and Tim Oates. “Time series classification from scratch with deep neural networks: A strong baseline”. In: *2017 International joint conference on neural networks (IJCNN)*. IEEE. 2017, pp. 1578–1585.
- [179] Philip B Weerakody et al. “A review of irregular time series data handling with gated recurrent neural networks”. In: *Neurocomputing* 441 (2021).
- [180] Jan Wessel. “Cycling in the dark—the impact of Standard Time and Daylight Saving Time on bicycle ridership”. In: *PNAS nexus* 1.1 (2022), pgab006.
- [181] Devon Paige Willis, Kevin Manaugh, and Ahmed El-Geneidy. “Uniquely satisfied: Exploring cyclist satisfaction”. In: *Transportation research part F: traffic psychology and behaviour* 18 (2013), pp. 136–147.
- [182] David H Wolpert. “Stacked generalization”. In: *Neural networks* 5.2 (1992).
- [183] 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.
- [184] Jianbo Yang et al. “Deep convolutional neural networks on multichannel time series for human activity recognition”. In: *IJCAI 2015*. 2015.
- [185] Shuochao Yao et al. “Deepsense: A unified deep learning framework for time-series mobile sensing data processing”. In: *TheWebConf 2017*. 2017.

- [186] William J Youden. “Index for rating diagnostic tests”. In: *Cancer* 3.1 (1950).
- [187] Baoding Zhou et al. “Smartphone-based road manhole cover detection and classification”. In: *Automation in Construction* 140 (2022), p. 104344.
- [188] Daniel Zwillinger and Stephen Kokoska. *CRC standard probability and statistics tables and formulae*. Crc Press, 1999.