CTU in Prague

Faculty of Transportation Sciences

Johannes Schlagheck

Generating Traveller Location Data from a Microsimulation Model

Master's Thesis

2016

Linköping University | Department of Science and Technology
Master thesis, 30 ECTS | Traffic Simulation
2016

Generating Traveller Location Data from a Microsimulation Model

- An approach to produce and analyze synthetic cellular connection records

Johannes Schlagheck

Linköpings universitet
SE-581 83 Linköping
+4613281000, www.liu.se

CZECH TECHNICAL UNIVERSITY IN PRAGUE
Faculty of Transportation Sciences
Dean's office
Konviktská 20, 110 00 Prague 1, Czech Republic

K620.....Department of Transport Telematics

MASTER'S THESIS ASSIGNMENT

(PROJECT, WORK OF ART)

Student's name and surname (including degrees):

Johannes Schlagheck

Code of study programme code and study field of the student:

N 3710 – IS – Intelligent Transport Systems

Theme title (in Czech): **Tvorba dat o poloze cestujících z mikrosimulačního modelu**

Theme title (in English): Generating Traveller Location Data from a Microsimulation Model

Guides for elaboration

During the elaboration of the master's thesis follow the outline below:

- Summarize the methods used traffic data collection with a focus on Mobile phone data
- Create the microsimulation model for the assigned region outside of Stockholm in AIMSUN
- Estimate the model parameters based on Sensoric traffic count data
- Extract vehicle trajectories from the simulation output
- Generate mobile phone network data (CDR) based on mobile phone cell coverage description and mobile phone usage model
- Propose the optimal usage of the data

Graphical work range: standard

Accompanying report length: ca 55 pages

Bibliography:

Michael Zilske, Kai Nagel, Studying the accuracy of demand generation from mobile phone trajectories with synthetic data, 2014, ANT-2014

Jameson L. Toole, Serdar Colak, The path most traveled: Travel demand estimation using big data resources, 2015

Master's thesis supervisor: **Clas Rydergren**

doc. Ing. Tomáš Tichý, Ph.D.

Date of master's thesis assignment: **July 30, 2015**
(date of the first assignment of this work, that has be minimum of 10 months before the deadline of the theses submission based on the standard duration of the study)

Date of master's thesis submission: **November 30, 2016**
a) date of first anticipated submission of the thesis based on the standard study duration and the recommended study time schedule
b) in case of postponing the submission of the thesis, next submission date results from the recommended time schedule

L.S.

doc. Ing. Pavel Hrubeš, Ph.D.
head of the Department of Transport Telematics

prof. Dr. Ing. Miroslav Svítek, dr. h. c.
dean of the faculty

I confirm assumption of master's thesis assignment.

Johannes Schlagheck
Student's name and signature

Prague .....June 1, 2016

Declaration

I hereby submit for the evaluation and defence the master's thesis elaborated at the CTU in Prague, Faculty of Transportation Sciences.

I have no relevant reason against using this schoolwork in the sense of § 60 of Act No. 121/2000 Coll. on Copyright and Rights Related to Copyright and on Amendment to Certain Acts (the Copyright Act)

I declare I have accomplished my final thesis by myself and I have named all the sources used in accordance with the Guideline on ethical preparation of university final theses.

Norrköping, 25.11.2016 Johannes Schlagheck

Abstract

This thesis contains an approach to do research on mobile connectivity data for the use in traffic modeling, while such data is not available yet. It describes the generation of synthetic Call detail records (CDR) from the vehicle trajectories of a microscopic simulation study. It investigates in how far it is possible to observe changing traffic conditions and route choices from these records. The simulation includes a highway stretch and residential roads in Solna, Stockholm and is carried out using the software Aimsun. The demand data is derived from sensors fixed along the highway stretch. A python script for the Aimsun advanced programming interface (API) is used to extract the vehicle trajectories from a running simulation. Mobile connectivity and call generation models translate the trajectories to CDR. Several data sets that differ in the underlying traffic demand and the grouping of regarded road stretches are generated. The data sets are analyzed in terms of total system load, average cell size, average cell dwell time and repeated connection sequences. Additionally, two ways to extract demand in origin-destination pairs from the data are compared. The first recognizes travel directions from the position of the subscribed cells and the second utilizes connection sequences. It is observed that it is possible to select traveling subscribers from the data by filtering them for large cells and connection patterns. Algorithms need to be trained to recognize those specific regional patterns that consist of cell sequences. Based on the load in these cells, changing demands in the network are identified. By analyzing the cell dwell time of subscribers, arising congestion in the simulated network is recognized quickly. It is concluded that route choices can effectively be identified by using connection patterns.

Acknowledgments

First, I want to express my gratitude to Clas Rydergren, who supported me from the very beginning of my work on this thesis. He helped me to find a topic and took a great effort to respect my own preferences regarding its detailed outline. For the time of more than one year he was quick to respond to any of my questions and always provided help in a way that showed his deep understanding of my work. First he did all this without any obligation and later continued as my official examiner. Next to him, Nikolaos Tsanakas stood by my side as my supervisor and patient lector of all my writing. I am grateful for his help with gathering input data and for all the helpful comments that greatly improved the quality of my report. Furthermore, I want to thank David Gundlegard for his voluntarily shared expertise in cellular networking. Special thanks go to Johanna Galarza Monta, who actively accompanied me during the work on this project. I cannot highlight enough, how important her enduring, daily support was to me.

Apart from being an academic work, this thesis also marks the completion of my education. In this context I would like to show my gratitude to all staff of Linköping university. Many times, they have aided me outstandingly and thus enabled my success in studying. However, most importantly I want to use this opportunity to thank Walburga and Berthold Schlagheck. Without their tireless support and continuous reassurance, the education that I was permitted to receive would not have been possible. I will be forever deeply grateful to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Traffic models are an important tool for engineers to estimate the current state of a road network. Based on them, experiments with new links or changed demands can be made. They also allow experiments on changes of the network before a decisions about investments have to be made. Modeling software includes advanced algorithms to compute realistic behavior of travelers. At the same time, it is possible to influence the behavior with many different variables. Mostly, the aim of setting a model's variables is to reproduce the traffic situation as closely to reality as possible. Therefore, precise and extensive knowledge of the traffic system is crucial. The knowledge is transferred into the model as input data. Without the right input, there can never be an appropriate representation of reality in the model. This would make the experiments performed with it and the conclusions drawn useless. For that reason, the collection of traffic data is a field as old as the centralized planning of road networks. Gathering data is a complicated topic, since traffic is the sum of the travel paths of many individual travelers. All of them have their own trip motivation, route choices and driving preferences, which results in a specific behavior. Thus, to be able to reproduce a realistic traffic situation, a high number of detailed data sets about travelers is necessary.

To include both, a high level of detail and a big sample size, is not completely possible. Throughout the time, different approaches have been developed to generate usable input data from the available sources. The classical way is to do surveys and to ask people from where to where they will travel each day. With the ongoing computerization of our civilization, other data sources became available that can be passively used as input for trip models. One example is the tax records of residents in the geographical research area. They contain information about the home and workplaces addresses as well as the number of cars per household. From that information, the most likely daily trip distribution can be computed and used as input data for traffic models.

This thesis focuses on a source for traffic data, that has become interesting due to the increasing use of cellphones. Along with massively increased spread of these devices, text messages and phone calls have become an important part of our communication within the last decade. Many people also communicate through their cellphones while

traveling. Additionally, cellphones receive text messages and periodically connect to the internet. For each of these tasks, the cellphone establishes a connection to the cellular network. For billing purposes, the network providers keep a record of all communication made with their customers' devices. This data is called call detail record (CDR). It contains a unique user id, a time stamp and the id of the cell, the cellphone is connected to. The cell id identifies, which base station is handling the connection. The location of the base station is known by the providers as well. Thus, a rough localization of a cellphone based on the cell ID in its CDR is possible. Furthermore, a CDR with several records for the same user can be used as a travel log. This makes it very interesting as input data for traffic models. It is not only cheaply available in a huge sample size, due to its simplicity the data can also be computed easily.

Research institutions are currently in a process of negotiation with network providers to gain accessibility to their network layout and their records. This process is slowed down by issues of privacy protection that forbid any third party use of user data completely. However, the process is on the way and the data is expected to be available in some form within the next years. To be able to make efficient use of it though, experience is needed with how to filter and interpret it. This thesis therefore suggests the use of synthetic call records to help understanding the correlations between traffic and CDR better. The data can be generated based on modeled vehicle trajectories. The precise locations of the base stations and thus the shape of the cellular network is not yet known. Nevertheless, there are publicly available location estimates of mobile phone cells. The thesis uses such public data for the generation of synthetic records. With these means a large data set that will be available for research purposes can be created with a low computational effort.

1.2 Aim

The aim of this research project is to evaluate the usability of mobile phone data as input for traffic models. While this mobile phone data is currently not available from the network providers, it has to be generated first. In order to support future projects, the methodology of the data generation is to be made reproducible. Therefore, the project intents to present a reusable way to generate CDR from a source that supports as many different road networks and traffic states as possible. Furthermore, the applied tools should be available to as many people as possible. This way, the methodology can be reused and improved by a wide range of researchers. The procedure is to be kept as easy and transparent as possible and at the same time lead to the generation of meaningful results. Several scenarios are to be investigated during the experiments. Based on the analysis of the different scenarios, conclusions about understanding CDR data as input for traffic models are drawn. This is crucial for being able to use the data effectively once it is available to researchers. The project specifically aims to answer the following research questions concerning the correlation of traffic situations and CDR:

- To what grade is it possible to distinguish a fast from a slow traveler in a synthetic CDR?
- How is a changing density in a traffic system visible in a synthetic CDR?
- In how far is it possible to distinguish travelers' origins and destinations or to identify specific route choices from synthetic CDR on a suburban scale?

All research questions target to improve the understanding of the correlation between road traffic and call records. Finally, the gained knowledge is intended to direct and motivate future research.

1.3 Methodology

The process towards the generation of synthetic CDR data in this thesis needs to be reusable, flexible and able to create meaningful results. These specifications imply compromises that have to be regarded when choosing methods and tools to proceed. A high flexibility of the methods is reached when they accept a wide range of different input. The wider this range is, the less the underlying model will be able to consider specific information that may be available in some cases. All data needs to be simplified to a bring it to a common denominator. At the same time, the quality of the output is decreased, the more the input data is simplified. Great care has to be taken find a satisfying compromise between flexibility through simplification and a high output quality. A similar trade-off occurs when the reusability is taken into account. By using tools that are publicly available, maybe even as open source, the process can be repeated by a high number of people. Thus it is preferable to apply such tools in general. Nevertheless, it is important not to rely on the compatibility of different software. The more programs that are involved in the process, the higher the risk that a future version of one of them will not work appropriately with the rest anymore. Thus, it would negatively influence the quality of the results. Finding a small number of specialized tools that are easily accessible can be a difficult task. The choice of methodology for this thesis is done bearing the three priorities reusability, flexibility and output quality in mind.

As a base for the data generation, a high number of vehicle trajectories need to be obtained. These trajectories can either be simulated or recorded from real vehicles. Chapter 2.3.1 describes ways to get a large number of real-world trajectories. Some providers offer such data almost worldwide and thus provide a large flexibility. Due to realistic route choices and demands, real trajectories have a positive impact on the quality of the results. However, the obstacles to buy the data each time it is needed makes the process less accessible. Furthermore, the project intends to avoid a violation of privacy by creating synthetic data. From an ethical perspective, it is better to avoid the use of personal location data in the first place. Thus, a simulation approach is preferred. Therefore, a microscopic simulation software is an appropriate tool. A traffic simulation software is specialized on generating a realistic representation of traffic. The underlying route choice models ensure a high entropy and due to the changeable parameters, different scenarios can be considered. Further motivation for the choice of a microscopic simulation approach is

given in chapter 2.1.1. Alternatively, manually generated routes from a navigation software could be used. By calculating a large number of routes, different scenarios could be examined. Geographical information software to do this is available as open source. However, it is not possible to realistically represent changes in traffic situations in this way. Furthermore, a study using this approach will always be more limited than with a microscopic simulation.

The microscopic simulation model used in this thesis exclusively includes motorized vehicle traffic. Since the input data does not include a separation of vehicle classes, all traffic is represented as private cars. Not including pedestrians, cyclists and stationary users in the model abstracts the results. They are not as divert as a real data set. In regard of the underlying research questions of the project, this simplification is accepted. At this stage of the research the focus lies on finding differences between the motorized travelers themselves. Additional information about the travelers, like their origins and destinations, is not used for the the result analysis either. The study intentionally narrows down the time of data collection and the geographical area. This way, the real time evaluation of connection records can be tested. An extension of the personal information to the travelers home or work places would simulate a knowledge that cannot be expected in reality. The simulation study is run over a time period of three hours from 6:30 until 9:30am with the averaged demand data of a regular workday. This time period includes the whole morning peak hour as well as the times around it with less demand. Choosing this time enables the simulation of different traffic states within the scope of a single simulation study. Additionally, the morning peak hour is of a great interest regarding demand and flow estimation. Thus it offers a good precondition to find meaningful answers to the research questions. The geographical research area needs to be one that is effected by the morning peak hour.

To represent a cellular network digitally, geographical data about the cell locations is needed. Additionally, as much meta data as possible can help representing the network realistically. Even though the network providers do not share such data, it is available from several sources. Decentralized data bases of old cellular network data are hosted by the government for some regions. These layouts are outdated enough, not to be considered company secrets any longer. However, they still contain realistic and detailed representation of typical cellular networks. More recent data can be obtained from online platforms that calculate the network layouts based on collected user data. A community of voluntary users saves and shares their mobile connection records through a smartphone application. Algorithms are applied on the raw data, to estimate the serving cells' location and meta data. The biggest community and thus the largest data base is provided by opencellid.org. Their network representation is chosen for this project. The big community generates a precise and up to date network representation through a large number of samples. It makes data with the same structure available for countries all around the world on an open source base. Thus the Open Cell ID data satisfies all main requirements of the project.

By correlating the vehicle trajectories with the cellular network layout, the cellular connection records are generated. The most commonly used and most available format

of such records are call detail records (CDR). This format is hence used as the structure of the experiment's results. To generate CDR from the input, a programming interface is needed that accepts input from a microscopic simulation and from Open Cell ID. This programming interface is found in the microscopic simulation software Aimsun. It offers a way to run scripts during an ongoing simulation. The scripts are thus enabled to extract data and to manipulate the simulation in real time. Additionally, Aimsun allows to import the Open Cell ID data as a spatial layer into a traffic model. It thus covers all the required tasks of the project. Handling many steps of the project within one software lowers the risk of incompatibility and hence increases the reusability. Figure 1.1 displays how the chosen methods interact in the experimental procedure.

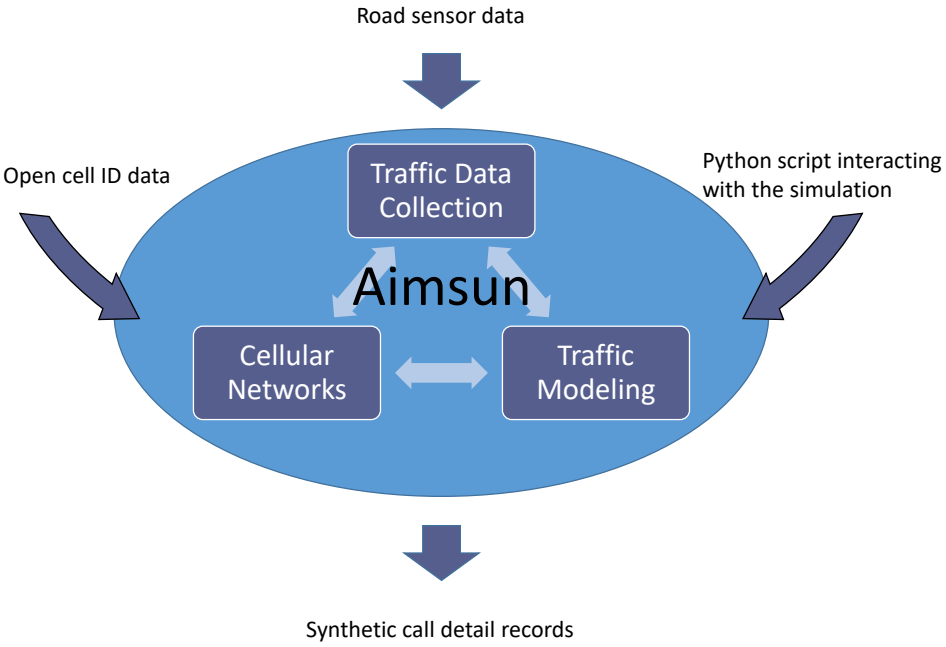

Figure 1.1: Interaction of methods in the experimental procedure

1.4 Limitations

This thesis is part of ongoing research in this wide and vivid field, the borders and interaction points need to be defined. This is important in order to place it within the scientific framework and identify its role in it.

The input used for the microsimulation model is highway sensor data. It is drawn from a database supplied by trafikverket in Stockholm. Previous projects have been done based on this data and helped understanding it as input for traffic models. Furthermore, these projects have generated tools to access the data base and extract clearly defined sets from it. These tools are being used to extract the input data for the traffic model in this project. Based on this input, the thesis conducts research on how to generate synthetic

CDR data. Data sets are created for one specific area using different traffic states. The project exclusively focuses on private car traffic. The generation of data for other modes of transport are left for future projects. One result of the thesis are the sample CDR. They can be used as input data to another micro-simulation model of the same area to try and match the results of the initial one. This is also left to do for future research. Within this thesis the generated data is analyzed on a basic level and some general conclusions from it are drawn. These conclusions are a result of the work as well.

On the side of mobile communication network modeling, the thesis relies on publicly available data, gathered by a large community of voluntary users. The applied network structure and radio resource management are as advanced as seen possible under the current conditions. Nevertheless, improvements on these models are subject to continuous projects with different scope and focus.

1.5 Outline

The generation of synthetic mobile phone data from a micro-simulation model requires the combination of different fields. A project with this extend and complexity needs a well formulated structure to remain understandable. This chapter describes structure of this thesis and thus helps the reader following its common theme. A visualization of the thesis's argumentative framework can be found in figure 1.2.

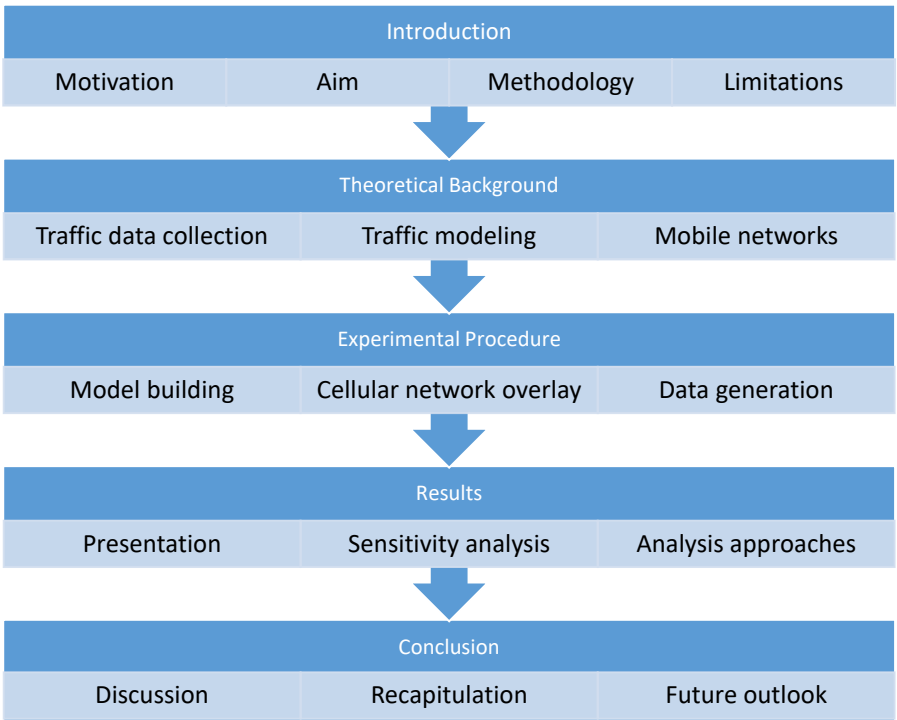

Figure 1.2: Outline

In chapter two, the **Theoretical Background** of the project is presented. The chapter hereby focuses on research specifically related to the topic and presents it in a brief and understandable way. Since the project intersects different research areas, this chapter consists of three pillars. The first of these pillars is about traffic modeling techniques. First it presents an introduction to the most important modeling approaches in traffic planning and therewith motivates the choice of microscopic simulation in the current case. The second section then describes the structure of microscopic traffic models. It is a preparation to understand the estimation of the model in the experimental part. A second major area of the thesis is mobile network design. The realistic representation of a mobile communication cell network has a big impact on the validity of the synthesized data. Therefore, it is important to understand the logic that modern cell networks are based on. This includes the distribution of cell towers and cell sizes as well as the administration of users that travel through the system. This section introduces the basic knowledge to understand how the subscription to the cell towers is decided in the experiment. The first two pillars, traffic modeling and cellular communication, intersect in the special approach of traffic data collection used in this thesis. The third section covers traffic data. It initially gives an overview of the currently and former applied data sources, and then focuses on the experienced use of CDR data in traffic modeling. On the one hand, this helps to understand what kind of data structures can be used as model input and how the CDR data may be translated to be applicable. On the other hand, it gives an understanding of the advantages and disadvantages of the new approach and points out how it can add value to the currently used ways.

Chapter three steps through the **experimental procedure**. It points out the single steps towards the generation of the data set. The three main parts of the experiment are explained. The first of them is to build a microscopic traffic traffic simulation model. This includes choosing an appropriate research area, representing this area in a software and finally choosing the correct input data for the different scenarios. As a second part, a representation of the cellular network is imported into the model and made usable during simulation studies. Once the model is set up, the third part of this chapter describes the process of generating the synthetic CDR data set from it. It includes a description of the Python extension in Aimsun that is used to extract and translate the vehicle trajectories. Further it explains how the theory of mobile network design and administration influences the way the cell distribution is performed. Additionally, a call likelihood model is introduced to simulate realistic mobile phone usage profiles.

Chapter four regards the work on the experiment's **results**. First the generated output is presented and its structure introduced. In the following analysis different approaches are evaluated. Some simply focus on the data sets as collection of records and analyze the system load or the number of records per vehicle. Other approaches utilize the common measures for cellular networks like the average dwell time or cell radius. Finally a more specialized learning algorithm is presented that relates connection patterns to both, specific route choices and traffic situations. Throughout the whole analysis procedure, three different scenarios with different demand backgrounds are applied. They are compared to evaluate the sensitivity of the output towards the change of conditions.

The **conclusion** of the project is presented in chapter five. In a first part, the results are discussed and summarized. The basic specifications of the project are recaptured and its performance put in relation to them. A review of the initial research questions recapitulates to what extend they have been answered by the experiments. The future outlook concludes what can be gained from the conducted research. Based on this, ongoing research projects are recommended and the possible scope of these projects described.

Chapter 2

Theoretical background

Understanding the project in detail, requires a certain level of background information about the topic. This information is presented in the following sections. Due to the high complexity, the information provided is largely focused on the current project. For more in depth information about the different fields, literature recommendations are provided in place. The thesis covers three research areas. Each of them is presented in one section. The topic of traffic data collection is central for the project. To understand the requirements to input data posed by traffic simulation, the modeling techniques are introduced first. Next, a section on cellular communication networks introduces the mobile phone records that are proposed as alternative input data. Both areas are intersected in the final section about traffic data collection.

2.1 Modelling of traffic systems

The first section frames the basics of traffic modeling and focuses on the microscopic simulation approach chosen for this project. The backgrounds of this modeling technique are presented to explain how it works and what requirements are to be regarded during the experimental procedure.

2.1.1 Simulation approaches

Traffic simulation is used to monitor current state traffic demand and predict future behavior. The discipline has its origin in the 1950s', when Lighthill and Whitham described traffic as an analogy for the flow of solid particles in a fluid in [42]. From then until now, in the face of constantly increasing demand on the road network, an efficient use and an appropriate extension of the existing network has become more and more crucial. Especially the introduction of computerized modeling tools has given the opportunity to simulate systems with high complexity.

The base of every simulation is a set of models. This mainly involves mathematical models that describe the connections between the acting parts of the system. Thus it is important to keep in mind the basic steps involved in modeling as can be seen in Figure

2.1. As Mitchell describes one of the principles in Operations Research in [46]: *'Model building implies making a statement of some or all the beliefs about the real world that the model builder thinks are relevant to the problem at hand. Using the model is viewed as the logical manipulation of these beliefs equally or more relevant to the problem'*. Thus it is vital to choose the most adequate model for each situation, since there is more than one way to model the same system. A clear description of the intention behind the modeling is a central part of this.

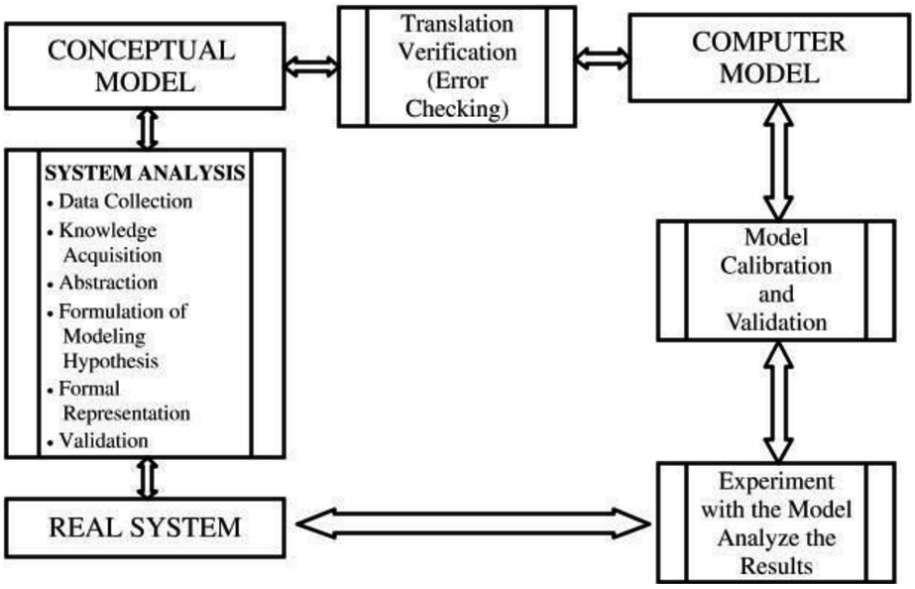

Figure 2.1: Logical steps of the model building process [6]

This section focuses on the appropriate choice of modeling technique. However it is important to keep the other arches of Figure 2.1 in mind, too. The result of a model can never be seen as a copy of the reality, but always has to be interpreted in an appropriate way. Further a well calibrated model for one research case might need to be adjusted to work for other projects too. When it comes to traffic simulation there are three mainly used strategies which are macro-, meso- and micro-simulation. For an overview of a recent state of the art in traffic simulation, [29] and [36] can be recommended.

Macro-simulation involves only the flow and speed on the monitored links. It regards the aggregated traffic on a link as one stream of travelers. A common analogy to describe how traffic is seen in these models is the flow of gases or liquids in motion as described in [60]. The mainly used mathematical parameters are speed, density, flow-rate and velocity. A macroscopic model may for example assume that the capacity of a multi-lane link is given by a fixed parameter and the speed of travelers on it will derive directly from the basic specifications. By simplifying the traffic on this high level, it is possible to model networks of a great extension without exceeding the limits of computational costs.

Meso-simulation involves a slightly higher level of detail. Still, single vehicles are neither distinguished nor traced. However, the groups that are regarded as one entity are chosen to be smaller in this case. To stick with the initial example, each lane could

be seen as one entity and the difference in density on the lanes are the trigger for lane changes of a group of vehicles. This way, the traffic is regarded more as the result of numerous individual decisions while still keeping the computational cost low.

Micro-simulation finally includes modeling of every single vehicle. This moves the focus of the model away from the specification of link parameters and speed-density functions. Microscopic models, rather rely on the precise reproduction of driver behavior. According to [9], this includes car-following, lane changing and gap acceptance models. Thus, the lane change example would be described as a chain of drivers' decisions in this case. Micro-simulation follows a much more natural approach to describe traffic. Nevertheless, such models are costly to compute and complicated to adjust. An even higher level of detail is regarded in **sub-micro-simulation** approaches. Those models even include the processes inside the vehicle, like driver-car interaction and driver distraction. In these models the required gap to perform a lane change could for example be influenced by the necessity of a change of gears. An introduction to sub-micro-simulation can be found in [29].

All given examples are mainly implemented to describe motorized road traffic. However, an extended use to describe pedestrian or bike traffic is possible as well. In that scope the models can even be applied to evaluate the functionality of escape plans in buildings or the capacity of a race track for a marathon event. For the research conducted in this thesis, micro-simulation is the appropriate tool. The objective to create CDR data from traveler routes requires the creation of personalized trajectories for those travelers. Thus only a microscopic simulation method can be valid. However, the level of detail of the trajectories does not need to be especially high. Cell triangulation, which is used for the positioning in these data sets is not that precise anyway. A sub-microscopic simulation approach hence is not necessary. To keep the computational costs as low as possible while providing the necessary output, micro-simulation is the tool of choice. In the following paragraph the underlying models of this simulation approach and its varieties are presented.

2.1.2 Models for microscopic simulation

A simulation of any scale is a combination of models, each of them describing parts of the behavior of the traffic system as a whole. The kind of the applied models depends on whether it is a macroscopic, mesoscopic or microscopic simulation. Within microscopic simulation the models focus on the interaction of single vehicles. Just as in a real traffic system, the sum of reactions of vehicles to each other results in the traffic state.

The core model used in a microscopic traffic simulation is the **car following model**. The basic assumption of this model is that speed and acceleration of a vehicle depend on the vehicle ahead. A driver's choice of speed is assumed to be determined by two objections. First to avoid a collision with the proceeding vehicle and second to travel at the driver's personally desired speed. It can be formulated in numerous ways like for example a mathematical equation, a set of rules or even fuzzy logic. A general version of a car following model can be written as:

$$Reaction(t+T) = sensitivity * stimulus(t) \tag{2.1}$$

Whereas t names the current time step and T denotes the following vehicles reaction time. The reaction within this model can be a deceleration or acceleration. This depends on the *stimulus* which typically can be the gap to a proceeding vehicle or the difference between current and desired travel speed. The *sensitivity* is a factor that determines the extend of the *reaction* [33]. The behavior and the complexity of a car following model is mainly influenced by its definition of these parameters. Advanced models include a classification of situations that cause adjustment of reaction and sensitivity. An example for this can be seen in Figure 2.2.

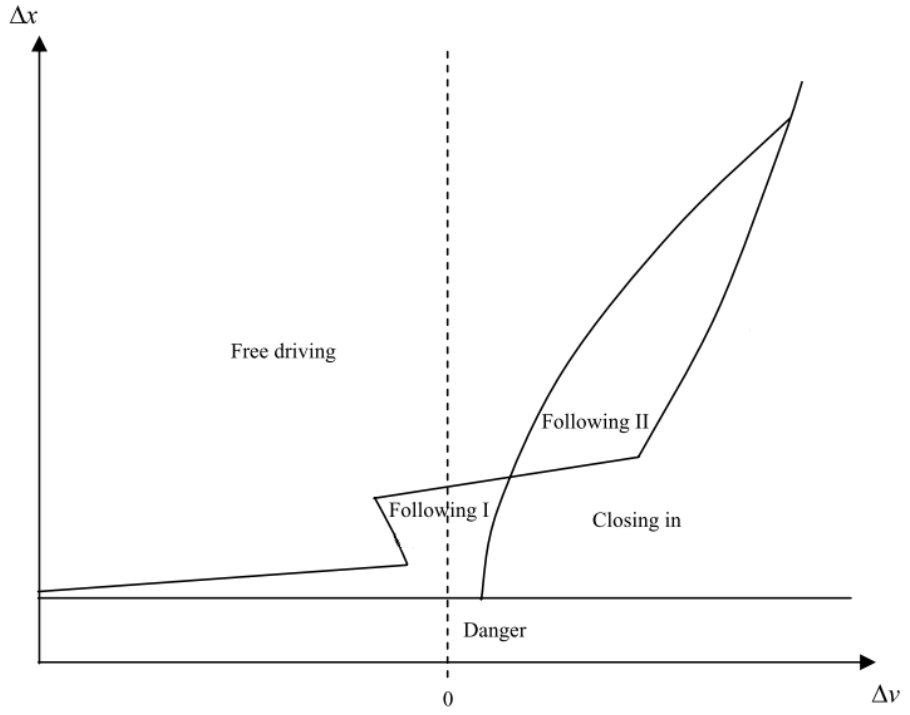

Figure 2.2: Classification of the Fritzsche car following model [23]

In the diagram, the choice of reaction is influenced by both the difference in speed and linear position of two vehicles. In combination these two values result in the theoretical time headway. It can be seen that everything undeneath a minimum distance is presumed as dangerous and will lead to more severe reactions than a "closing in" situation with more distance between vehicles. In the *Free driving* area the proceeding vehicle is of little importance to the follower, thus the sensitivity factor is close to zero. An overview of popular car following models can be found in [14] and [50] give a comparison of their specific behavior.

The micro-simulation software Aimsun that is used in this thesis relies on a car following model based on the one introduced by Gipps in 1981 [26]. Despite being more than 30 years old, it is still one of the widely used models for computerized traffic simulation. Its major advantages are the clear physical reproduction of drivers' attempts to

"smoothly reach the desired speed or to safely proceed behind his leader", as formulated by [19].

Another model, used within microscopic simulation is the **gap acceptance model**. This model is used whenever a vehicle attempts to join an existing traffic stream and thus is waiting for an appropriate gap. The first major situation, when the model is applied, are give way intersections including roundabouts. A car waiting to enter will make the decision to enter or not depending on the gap to the next upcoming car. Gaps are typically calculated in the predicted time until the crossing vehicle reaches the intersection. [2] presents and compares several kinds of gap acceptance models. A second common application of these models are highway mergings and lane changing. In these cases, the classic gap acceptance models often fail to reproduce realistic driver behaviour. On an acceleration lane for example, drivers tempt to accept smaller gaps and greater changes in their own speed in order to enter a highway. To represent this in the model, some parameters have to be adjusted, as proposed in [35].

On the contrary to the previously presented models, the **route choice model** does not directly affect the traffic flow. It merely distinguishes the path, a single vehicle chooses to get from it's origin to the destination. The major parameter distinguishing these models is whether they are static or dynamic. According to [10], static models work in two steps. First they distinguish a set of possible routes that connect the traveler's origin and destination. Second, they rate every possibility based on its costs. The costs are a fictional value, generated by a formula that uses the routes' given parameters. In combination with the traveler-specific preferences (e.g. value of time) the most appropriate alternative is chosen. This process is repeated for every vehicle, entering the simulation area. On the contrary, dynamic models add a third step at the end of each loop that updates the routes' parameters based on the changed traffic load. This influences the route choice of following vehicles. Some dynamic models even update the route choice due to changed traffic conditions, while a vehicle is in the simulation.

The formulation of route choice models can be classically deterministic with consideration of cost minimization and entropy maximization alike. Alternatively, a logit model approaches are popular, since they model human decision making especially well. [39] presents advantages and disadvantages of such models and suggests C-Logit nesting of similar routes as an improvement. An extensive overview of route choice models can be found in [12].

2.2 Cellular networks

The second section gives an overview of mobile communication network design and its functionality. First, a brief introduction of basic cell layouts is given. The spotlight in this section is put on the signaling data that occurs during wireless communication, since it will be reproduced as a result of the experiments. Further, the network specifications for traveling users are presented to motivate the later applied algorithms regarding cell changes.

2.2.1 Cell structures based on population density

Mobile communication has become a part of everyday life all around the world. The market is constantly growing and replaces more and more of the traditional wired connection. Especially in developing countries intensive mobile phone use has become part of the culture. Next to voice and text communication, internet access has become one of the key functions of the networks. As the demands for velocity and capacity grows, new network generations have been implemented to cover more and more traffic. Chapter 1.3 of [25] provides an overview of the development.

As a first globalized network, the global system of mobile communication (GSM) defined common standards that enabled unbounded roaming. Although the more advanced network types like 3G and 4G gain importance as mobile internet demand grows, GSM systems still offer the backbone of every mobile network. They contain the biggest infrastructure and ensure basic provision in almost every populated area around the world. GSM systems are found operating in frequency bands around 900 MHz, 1.8 GHz or 1.9 GHz. The lower frequency bands serve as macro cells with large size and high power transmission towers, whereas the higher bands are operating as micro cells [69]. Macro cells are used to offer a wide range coverage in rural area without much traffic. There, one single cell can reach a cell radius of up to 35km. So-called mega cells are offering basic coverage in very remote areas and can even reach radii up to 500km. In urban areas, macro cells are also used for travelling subscribers. This avoids an unnecessary high number of handovers and thus lowers the risk of a dropped call. Micro cells in the higher frequency bands can exist parallel with the others and provide extra capacity for stationary users. Their cell radius typically does not exceed 1km as is stated in [70].

Mobile phone networks that are based on coverage or capacity also differ in the shape of their cells as can be seen in figure 2.3. The first kind uses one omni-directional antenna to cover a circle around the base station. In this way, one frequency is used for the whole area and only a low number of base stations is needed. For large cells this layout provides the best signal strength even at the cell borders. Capacity oriented cells usually work with sectored antennas as can be seen in the right half of figure 2.3 that shows an example of a 3-sector antenna. Here, one base station includes several cells that each cover 120° of a circle. This way, every base station can use multiple frequencies and hence offer a higher capacity. Additionally, a lower transmission power can be used which is good for the mobile stations' battery lifetimes [56]. A major advantage is that the shape of the cells is better adjusted to an urban street pattern what avoids shadowing effects from buildings. Actually, the shadowing can in this case be used to avoid interference with other cells. More information about cell layouts can be found in chapter 5.5 of [25] and in [21].

How well a signal can be transmitted within different areas is described by propagation models. An overview of them can be found in Chapter 3.9 of [25]. One of the widely used models is the Stanford University Interim model implemented by the Institute of Electrical and Electronic Engineers (IEEE) presented in [55]. It is designed for the GSM frequency bands between 800 and 1900MHz and includes three different parameter sets

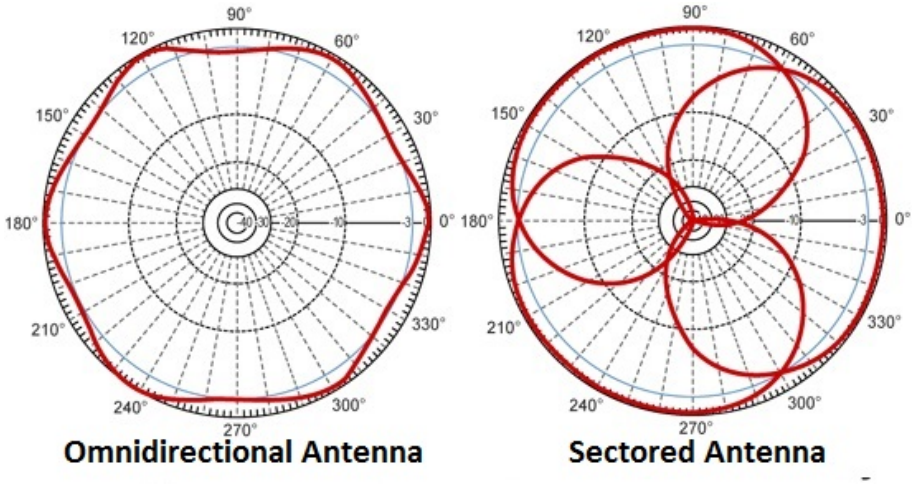

Figure 2.3: Transmission spectrum of mobile phone antennas [37]

for urban areas, suburban areas and open terrain. The models structure is based on the physical boundaries of radio wave propagation, whereas its parameters have been estimated empirically. Propagation models play an important role in establishing handover algorithms and in designing the layout of the overall cell structure of a network. They can help deciding where to put physical cell borders and how to position the antennas to reach maximum coverage.

2.2.2 Signaling data in cellular networks

Wireless connection in a cellular system consists of a standardized communication between the network's base station controller (BSC) and the mobile unit. [64] provides several examples of the two-way connection. For example, the signaling during the mobile unit's start up process or while a call is being placed. In addition to this, the two counterparts frequently exchange measurements to verify the mobile unit's location or to adjust the cellular power level. Exemplary, table 2.1 shows an extract from the communication during a handover process. Apparent is the Change in cell-ID in the beginning of the process and the message "Handover Complete" in the end. The process is supported by continuous signal measurements of both sides. Depending on the network protocols, the mobile unit can monitor several base stations at the same time. The data has been captured by using the TEMS investigation tool by Ericsson. A summary of all signaling information exchange that is covered by the software can be found in [8]. The most important measures of a mobile network are the received signal strength (RxLev)/, the received signal quality (RxQual) and the timing advance (TA).

This kind of in-depth data collection just happens for diagnostic purposes. Normally, only some summarizing data sets are being stored by the network providers. As listed by [71] these sets contain the call detail records (CDR), cell change updates (CCU) and network measurement reports (NMR).

CDR data is stored by the provider for billing the customers. Hence, it contains all

Table 2.1: Signaling data collected during a handover

Time	MS	Direction	Message Type	All-Cell Id	All-BSIC)	All-RxLev Sub	All-RxQual Sub	All-TA	All-MS Power	Distance	Dwell time
11:59:08.93	MS1	DL	Channel Information	29090	16	-84	0	1	30	71,63	17,39
11:59:08.95	MS1	DL	I-CMD	29090	16	-84	0	1	30	71,63	17,40
11:59:08.95	MS1	DL	Handover Command	28001	25	-84				71,63	17,40
11:59:08.96	MS1	UL	Handover Access	28001	25	-84				71,63	17,41
11:59:08.96	MS1	Internal	MPH State Report	28001	25	-84				71,63	17,41
11:59:09.04	MS1	Internal	PH Block Report UL	28001	25	-84				71,63	17,86
11:59:09.04	MS1	Internal	PH Block Report DL	28001	25	-84				71,63	17,86
11:59:09.04	MS1	Internal	MPH Channel	28001	25	-84				71,63	17,86
11:59:09.04	MS1	Internal	RR State Report	28001	25	-84				71,63	17,86
11:59:09.07	MS1			28001	25	-84				71,63	17,87
11:59:09.07	MS1	Internal	PH Block Report UL	28001	25	-84				71,63	17,87
11:59:09.07	MS1	DL	Channel Information	28001	25	-84				71,63	17,87
11:59:09.07	MS1	DL	Physical Information	28001	25	-84				71,63	17,87
11:59:09.07	MS1	Internal	MPH State Report	28001	25	-84				71,63	17,87
11:59:09.17	MS1	Internal	Channel Mode Report	28001	25	-84				71,63	17,93
11:59:09.17	MS1	UL	Handover Complete	28001	25	-84				0,00	0,00
11:59:09.18	MS1	Internal	RR State Report	28001	25	-84				0,00	0,01

information about established connections, may it be for internet access, texting or a call. It contains an anonymous user ID as well as the ID of the serving cell and a time stamp. Handovers that happen during an active call are not included in the CDR. The cell ID column only contains the cell that initially established the connection. The **CCU** is used by the provider in order to keep track of mobile units within the network. This is crucial to keep the costs low, when the device has to be contacted by a base station. In most networks, several cells are combined as location areas. In this way, there is no update needed for every cell change, but only when entering a new Location Area. The size of these areas is a trade off between the signaling costs for one update and for finding the phone when needed. Due to this, the cell data from the CCU is not as precise as in the CDR. However, it also contains idle users, which offers a much larger sample size. Idle mobile units that don't move across cell borders are still listed in the **NMR**. This data set is generated by frequently exchanged measures for important network variables, like they were listed before. It is only collected occasionally, but offers the most detailed information. On the contrary to CDR, CCU and NMR are not stored in specific data bases, what makes them harder to access. Thus, it is the most realistic approach to focus on CDR, when using signaling data records for extended purposes.

2.2.3 Network design for mobile users

Traveling users pose specific challenges to mobile communication networks. For them, big cells have to offer coverage over long distances, while at the same time the network has to offer capacity. Further the inevitable scenario of a cell change during a connection is a demanding procedure for the radio resource management. A general evaluation regarding these effects and their related parameters is given in [73]. One important part of designing networks appropriately for mobile users is to organize multiple neighboring cells in one location area. In this way, the paging costs can be balanced against the call receiving costs. The location areas' borders are to be chosen in a way that the intra area traffic is much bigger than the inter area one [70]. However, it is not only the minimization of costs that has to be regarded. Every time an active mobile units crosses a cell border, the connection has to be reestablished to a new base station. As [48] states: "The major parameter in any network is defined by its Quality of Service (QOS) and

handoff decision scheme plays a major role in QOS." To create a seamless handover from one cell to the other, handover algorithms follow specific rules to decide whether to stay connected to the previous or switch to a new cell. Their decisions are supported by the constantly taken measures of either the base station, the mobile unit or both.

The complexity of a handover algorithms task can be seen by the large number of different approaches that are continuously developed and improved by researchers. Related information can be found in chapter 3 of [7] and in [3]. Most Handover algorithms mainly use the received signal strength and the received signal quality as measures. These algorithms typically have to handle a trade-off between a fast handover decision and a desirably low number of handovers. The longer it takes to make the handover decision, the lower the received signal strength becomes before it is performed. This lowers the call quality and even increases the risk of a dropped call. On the other hand, [43] describes the risk of a so called ping-pong effect for too quickly taken handover decisions. This means that multiple handovers between the same cells in a short time can occur only due to signal fluctuation. According to [45] a maximum threshold of signal strength or quality is a first step to avoid unnecessary handovers. In combination with a minimum hysteresis between the values of two different cells, ping-pong effect and handover decision delay can be effectively balanced.

Ongoing research in this topic mainly focuses on the utilization of positioning data, like done in [34] and [44]. In this way, algorithms can learn from the effects of past decisions taken for users in the same places. Over time, cell borders can be developed that help to decrease the handover decision delay significantly. A crucial part in these algorithms however is the availability of precise location data that can be hard to obtain in traditional GSM networks.

Another dimension to handover algorithms is added by the modern multi layer networks. A typical structure for those can be seen in [24]. For different kinds of users, multiple cells are available at the same place. Each of them designed to serve for specific purposes. Next to the previously regarded horizontal handovers, also vertical handovers can be performed. This term for example, describes the migration from a macro to a micro cell and is based on different decision parameters as described in [8]. Vertical handover algorithms need to distinguish the user's purposes to find the best fitting cell layer for his demands. One approach to do so is presented in [53] and focuses on the residual time as most important indicator.

2.3 Data collection

Collecting traffic data of high quality in a large scale is the far side aim of this and ongoing research projects. The third section of this chapter is about traffic data collection in all its forms. The focus lies on the advantages and disadvantages of common methods as well as the use of mobile connection records. It is aimed to help the reader understand all crucial points in this topic and the opportunities that are connected with the new kind

of data.

2.3.1 State of the art in traffic data collection

This thesis aims to generate mobile phone connection data in order to train computerized models to use it as an input. To understand the opportunities offered by this procedure, however it is important to see where the advantages in comparison with other data gathering techniques lie. This chapter provides an overview of classic data collection techniques including their advantages and disadvantages.

One of the oldest, but still applied techniques is the vehicle owner interview. Questionnaires are given out to inhabitants of a research area and they're asked about travel specific details. These procedures are rather labour intensive and costly per dataset obtained. Furthermore, they are not popular amongst participants, since they require a lot of effort from them. Common procedures are mailed questionnaires, telephone interviews or even face to face interview and travel diaries. All techniques suffer from a typically low response rate of less than 60%. Nevertheless, they can deliver in depth information that pure traffic counts cannot. This includes trip purposes, car ownership rates and frequent habits according to [13]. By choosing samples from different demographic, racial and social groups, even rather small data sets can be used to calculate a representation of the whole population. Maybe the biggest disadvantage of this method however, is the long friction between the collection and the availability of the data. It typically takes several years until a nationwide travel survey is evaluated and its results can be utilized.

Stationary road sensors are another classic technique of traffic data collection. Within this group, many different technologies are regarded and continuously being developed. Some of the most important road sensor technologies are be described here. A more complete list can be found in [5]. A group of sensors detects the passing vehicles based on the pressure they put on the road's surface. Amongst them, bending plates, pneumatic road tubes and piezo-electric sensors are popular. Since a vehicle's weight is carried by its axles, all these sensors are regarded as axle sensing technologies. Next to the pure number of axles, they can also measure the weight passing over them. Based on these two measures a vehicle classification is possible as it can be found in [22]. Presence sensing techniques on the other hand, only detect the existence of a vehicle. Common technologies are inductive loops and magnetic sensors. These devices sense cars based on the metal they are made of. They can only differentiate based on the length, which is not as meaningful as the advanced axles sensing classification. As road sensors are commonly applied in many road networks there is a lot of experience regarding their usage. Further their constant presence at the same place allows highly comparable data collection over long time periods. Their fixed position on the other hand also makes them inflexible. An additional issue, especially for the weight sensitive axle sensing technologies is their vulnerability. This is an especially sensitive disadvantage regarding the large effort it requires to replace or repair them.

Roadside cameras offer a more flexible alternative to static sensors. Since they are not embedded in the roads' surface, temporary installation is easier. Sometimes they are also referred to as non-intrusive road sensors. Next to film cameras, radar, ultrasonic or infrared technology can be used. In combination with a software, reliable traffic counts can be performed. Modern software cannot only differentiate between different modes by measuring the extent of a moving object. By recognizing and reading a car's license plate, vehicles can even be tracked along their path as stated in [58]. For studies in greater detail, lane wise counts and even vehicle occupancy as done in [18] is possible too. The latter is especially interesting for the enforcement of the correct use of carpooling lanes. However, the required cameras get vastly expensive, if detection at high speed is required. The camereas are easlily getting confused by pets, dummies or large pictures in the seats. Newly developed cameras avoid this weakness with advanced image taking techniques. The detection is done by taking pictures while exposing the car to short wave infrared light. The light is reflected by iron carrying blood cells and thus detects humans. Recent trials described in [15] conclude that the technology only works at 87% efficiency so far and especially still has issues with back seat detection.

Bluetooth and wi-fi devices are nowadays present in many cars and on pedestrians. Most common are smartphones running on one of the two big operating system iOS or Android. These devices are as default set to scan continuously for available networks and thus can be identified by their MAC (Media Access Control) address. The address is unique for every device and enables the device to be re identified reliably. This enables the creation of movement patterns of users by installing detectors in a certain area. To detect a device with this technique, it does not have to be in use, only the wi-fi or bluetooth connection has to be activated. An example for the commercial application of this technique is provided by [41]. The detectors are rather cheap. High speed identification however, does not work reliably. Currently at a speed of 100 km/h an average of 80% of the devices is detected. A precise location can be achieved by measuring the received signal strength of the devices, which gives an estimate of their distance to the detector.

All the traffic data collection methods presented so far rely on stationary detectors that recognize and identify bypassing vehicles in some way. Thus, the obtained data will be a spot study of the traffic state at one specific position. An alternative approach is given by global positioning system (GPS) devices carried on board of vehicles. These devices continuously elaborate a vehicle's current location based on satellite signals. This way, detailed trajectories, not limited by the position of roadside sensors can be obtained. The principle is known as floating car data collection and described by [54]. A big advantage of this approach is the freely usable satellite infrastructure with a high precision in positioning. [38] estimates the location error as typically less than 30m. Furthermore, the receivers are rather cheap and already installed in many vehicles as navigation systems or for fleet management. Few vehicles however, are equipped with technology to forward the GPS data. This is not done automatically, since GPS works with a signal being broadcasted by the satellites and simply received by the on board unit. Additional technology is necessary to share the location data such as portable hard drives or general package radio service (GPRS) uplinks. The second option is the only possibility to gather real time floating car data in this way. GPRS makes it possible to transfer data packeges

over cellular networks and thus enables uploads from almost any place. A chance to get a high number of data samples especially in big cities is the cooperation with taxi companies. They often have their vehicles equipped to allow efficient ride distribution and driver monitoring. The rate of equipped vehicles compared with the whole traffic load will always be low using this technique. Based on the taxis' speed however, conclusions about the current density on the links are possible. An example for traffic control based on this data can be found in [32].

2.3.2 CDR data in traffic modeling

Traffic models are an essential tool for urban planners and transportation engineers to plan and evaluate necessary investments in infrastructure. Since every project requires a different focus on details or totality, the models can be shaped in many different ways. Especially for large scale models that are used to evaluate the traffic inside a whole region or city, origin-destination (OD) matrices are of great importance. These matrices show how frequently journeys have been started or finished in one place. Hereby the focus is set on trips from and to work. Those are expected to be frequent and to have an impact on the morning rush hour. For the data to be meaningful, the monitored notes have to be of an appropriate size. Most important it is however, to provide a high number of observed trips to achieve a good representation of the real commuting situation. This is a crucial point in traffic modelling, since it is generally hard to grasp satifying input data. A classical way of gathering it is to perform a household survey. This way, a high detail level for relevant trips can be achieved. However, surveys are also expensive, slow to be evaluated and only offer a small data sample. Another regularly used way of data collection are roadside counts. These can either be performed by persons or by automatic sensors mounted in the pavement. Here again the collection is quite costly and can only be done on a limited number of links [52].

An alternative way of input data collection could be provided by a recently developed method. The use of passively collected mobile phone connection data can be utilized to create a movement record of the users. Mobile phones connect to their provider's antennas when a call is made, a message is sent or the internet is used. The collected data is called call detail record (CDR) and is used and saved by the provider for billing. Typically, it contains a timestamp for each connection establishment of a cell phone, a unique but anonymized caller ID and the cell phones current location in latitude and longitude. The position is estimated by using either standard triangulation algorithms or the serving cell towers locations as stated in [4]. A sample of such data can be found in [31] and its specifications are further explained in chapter 2.2.2. Extracting regular travel routes from these records is a comparably cheap way to obtain an OD matrix. Thanks to the popularity of cell phones, a big and representative data base can be achieved. [66] sees opportunities especially in developing countries, where more than 90% of the travellers carry a phone. Thus it can be a great alternative to installing costly roadside sensors in those countries. There have already been several successful approaches using CDR type data in real world OD estimation. Examples can be found in [16],[67] and [59]. In [30] the regions passed along the travelers' way were evaluated as well and have been

added as extension of the classical OD matrix. Commercial projects to utilize the data are currently exclusively existent in the United States, since there the privacy laws allow the spread of anonymized mobile phone records. Several communities have obtained internal and external trip matrices. In Moore County, North Carolina for example, [57] uses CDR to differentiate between local and non-local travelers on a stretch of highway running through the county. As reported by [17] the same company, contracted for this project was also most recently hired to monitor the movement of people related to the 50th super bowl match in San Jose, California.

Despite of all promising research, it has to be kept in mind that nobile phone records have originally not been intended for the use of movement monitoring. The phones location is just collected as a side product of its communication with the base tranceiver station. Thus it is not focussed on accuracy and an adequate localization precision has to be assumed. On the one hand in order to get as good results as possible and on the other hand, to avoid the generation of fake trips due to imprecise localization as discussed in [31]. A typically achievable localization error using the classical localization approaches is between 200 and 300 m. However, [20] argues that the increased spread of mobile phones including a GPS receiver could lower this error in the future significantly. Furthermore, the usage of the data requires assumptions like defining what to interpret as work or homestay. This is for example done by the commercial provider of mobile phone record based commuting studies AirSage in [1]. For them a regular extended stay at on location during the night defines one person's home and a likewise pattern during the day defines a work place.

CDR data offers a big chance to obtain more recent trip data and a higher number of probes to estimate OD matrices with comparably low costs. The method has already been validated by several studies and its results hold in a comparison to traditionally estimated models. Additionally, know-how in analysing CDR data can be applied for more than just the creation of traffic models. As an example, [11] describes an application that can recommend extensions to a city's public transport network based on CDR data automatically. Alternatively there has been research about predicting the next outbreak of a disease based on CDR data in [65]. This shows how widely CDR data can be used. Especially in a future perspective, when an increased precision can be expected and experience about its application has been gained, this data offers big opportunities. To stick with an example of traffic planning, [27] shows how GPS tracks can automatically be analyzed to identify the mode of transport used. A similar approach on a bigger scale is possible with CDR data too, once the precision is improved.

Nowadays there are still some obstacles to overcome, before the data's full potential can be utilized. In many parts of the world including Europe, CDR data cannot be used yet, since the privacy protection of the mobile phone users is not solved convincingly. The policy and law about privacy rights differ from continent to continent. An overview about the current state is given in [40]. However, there will always be some effort have to be put in the anonymization of the records before they're ready to be used. Since user mobility patterns are rather specific, even slightly anonymized data can often be allocated again. [47] shows that the use of just anonymized caller IDs is not enough to avoid the

identification of specific subscribers in many cases. Further, the initial purpose of mobile phone records is neither the users' precise localizations nor the use as travel log. Hence, there will always be noise that has to be filtered out and assumptions regarding the trips to be made. A basic example for this is the expected rate of travellers actually carrying a mobile phone that is turned on as stated in [72]. On the contrary to classical travel surveys, the motivations for trips and the used modes are also not included in the raw data. This requires further assumptions.

Chapter 3

Experimental procedure

The theoretical knowledge presented previously is put to use during the experimental procedure in this chapter. It leads step by step from initializing a microscopic simulation model, preparing the demand data and applying the mobile connectivity model on it. A lot of attention is also paid on the details of the script to generate the connection records. The description is aimed to explain the details of the experiment and to help understand its specifications. The basic experimental process is summarized in figure 1.1 in section 1.3. It visualizes the procedure described in the text step by step.

3.1 The microscopic simulation model

The data generated during the experiment can only be as good as the model at use. Hence, the implementation of the microscopic simulation in Aimsun is an important part of the procedure. This section guides the reader through all parts of the model's setup and motivates important decisions like the choice of the research area and of input data.

3.1.1 Choice of the geographical research area

The choice of a geographical area for the research has a major impact on the utility of the generated results. The target of the project is to perform research on the effects of traffic situation on CDR data sets and to evaluate their usability to determine route choices. Therefore, the geographical area of choice needs to fulfill certain criteria in order to enable the generation of valuable results. This section names this criteria and therewith motivates the choice of a specific region to work with.

First, a large number of cells is needed to have the chance to follow the travelers along their path. This can be achieved by either simulating a large network or one in a region with small cell sizes. Simulating large areas is troublesome in microscopic simulation studies, since the model's complexity grows fast in a bigger network. A compromise to avoid this conflict is to include a highway in an urban area. Due to the limited number of on- and off-ramps, the model does not become too complex for a longer stretch of highway. At the same time, the average cell size in urban areas is smaller, since the network is

designed for high capacity instead of large area coverage. Another advantage of choosing an urban area, is a better network coverage. It does not make sense to pick a region where cell phones are without reception regularly. In terms of utility of the generated data set, small cell sizes help increasing the precision of the localization. A multi-layer network with macro and micro cells helps estimating a travelers speed based on their connection record. Both these prerequisites are mainly present in urban areas as well. To include these effects in the model, it further is important to regard a wide range of road classes from small residential streets to highways. Finally, the availability of input data is crucial. The better the input data is, the closer the model's results can represent the reality.

Following these requirements, the neighborhood Solna in the north of Stockholm is chosen. A map of the region can be seen in figure 3.1. The suburban region contains a stretch of the highway E4, connecting the center of Stockholm and Arlanda airport. It is one of the main routes for commuters in the greater region of Stockholm. The highway is equipped with a high number of road sensors to count the vehicle flow once per minute. At the same time, a relatively high number of exits results in a parallel arterial road with a lower speed limit. The whole geographical area is fenced by train tracks in the west and the lake Brunnsviken in the east. This keeps the number of origins and destination for the model low. The neighborhood Solna includes urban roads with speed limits of 30 or 50 kilometers per hour. These roads offer a contrast to the highway section. Several dense residential buildings in the neighborhood generate demand for its streets. The extend that is shown in figure 3.1 covers the greatest part of the geographical areaof research. Only the E4 highway is continued for 2 more kilometers in the north. This highway stretch does not include any additional exits. Only roads that can be used by motorized traffic are part of the model. Pedestrian areas that can be seen on the map are hence excluded from it.

The chosen geographical area includes the main requirements for the research in this project. The different road classes with their distinct traffic structure offer a base for evaluation regarding the research question in how far it is possible to filter CDR data. The numerous route choice possibilities are crucial for examining in how far an OD estimation is possible under real conditions. The E4 highway bypassing Solna is an important connection of Stockholm and its surroundings. It thus is hugely affected by the morning rush hour. This ensures a wide range of traffic conditions within the simulation time frame. an important precondition to examining in how far a changes in traffic are apparent in CDR data. Furthermore, due to Solna's natural limitation, the model can be kept relatively simple. This enables the microscopic simulation of such a large area, without raising the complexity too much. A trustworthy base of input data is given by the sensors on the highway stretch. As part of the densest metropolitan area in Sweden, Solna is of major interest for traffic planners and thus its choice makes the projects results more valuable.

Figure 3.1: Map of Solna in the north of Stockholm [28]

3.1.2 Building the model

The first step of the experimental procedure is to build the microscopic simulation model. For this task, the traffic simulation software Aimsun of the Spanish company TSS is used. The software offers a wide range of modeling functions that are helpful for the project, including an easily understandable environment for network building. Further, the implementation of a two way system for user built applications is crucial. Such an application will later be created in order to create the CDR data set from the simulated vehicle trajectories.

As a basic part of traffic modeling, the road network has to be rebuilt within the software. The representation has to be done in a way that represents the real area in the necessary level of detail, to use the input data effectively and to obtain meaningful results. At the same time, due to the limitation of computational resources simplifications have to be made. At this point it is of importance to keep the model's purpose in mind.

Since it cannot perfectly recreate the whole environment, the focus has to be on those parts that matter most regarding the output. In the current case these are:

- Differentiating between road classes
- Covering a large area
- Creating free flow and congested traffic
- Including route choices
- Simulate commuting trips during the morning rush hour

In Aimsun road stretches are called **sections** and created by dragging and dropping them across the screen. To rebuilt an existing traffic system, a background map is needed. In this case, since the trajectories of the model are to be exported, the background map has to include correct geographical information. In the project a sample network provided by the Stockholm traffic authority is used. It represents each road as a poly-line in an Esri shape file, a format that can be imported into Aimsun. Hereby it is important to use a proper map projection technique that matches with the one used for the cell towers. For this project the projection "WGS84" is used.

By default, all modeled roads have the same meta data. To adjust parameters like the number of lanes, speed limit or capacity the program offers road classes. Most information is drawn from the background map. The sections are modeled as four different road classes including freeway, on-/off ramp, arterial and urban street. For more detailed information about the shape of intersections and links, online Satellite pictures from Google are used. However, in some cases the network's shape is adjusted due to the requirements of the simulation software. The simulated vehicles struggle for example with highway on-ramps without acceleration lanes. These are added to mirror the correct driving behavior rather than the exact layout. Due to simplicity, all urban **intersections** are modeled non-signalized. Either they are designed as roundabouts or as *yellow-box intersections*. These are described in [63] as follows:

"A vehicle approaching a Yellow Box Junction will avoid entering the junction area whenever the preceding vehicle is moving at a speed below a specifically set speed"

The intersections can well be seen in figure 3.2. The figure also shows how sections are modeled in Aimsun as one way roads. Even small residential streets have to be built as separate sections for each direction. Thus overtaking on these roads is not possible in the model. Parking along the road is not regarded in the model either. Traffic is exclusively originated and destinated in centroids. In general the layout of roads within Solna is simplified. This simplification helps avoiding some small intersections and thus lowers complexity. These intersection are crossing paths for pedestrians or streets only used for parking. Since parking on the road is not regarded in Aimsun, it represents reality closer to replace the intersections by centroids. Some of these streets generate so little traffic that they have absolutely no effect on the models results.

Centroids are represented in Aimsun by circles that are connected to loose ends of sections as can be seen in figure 3.2. One centroid can be used as source and sink for any

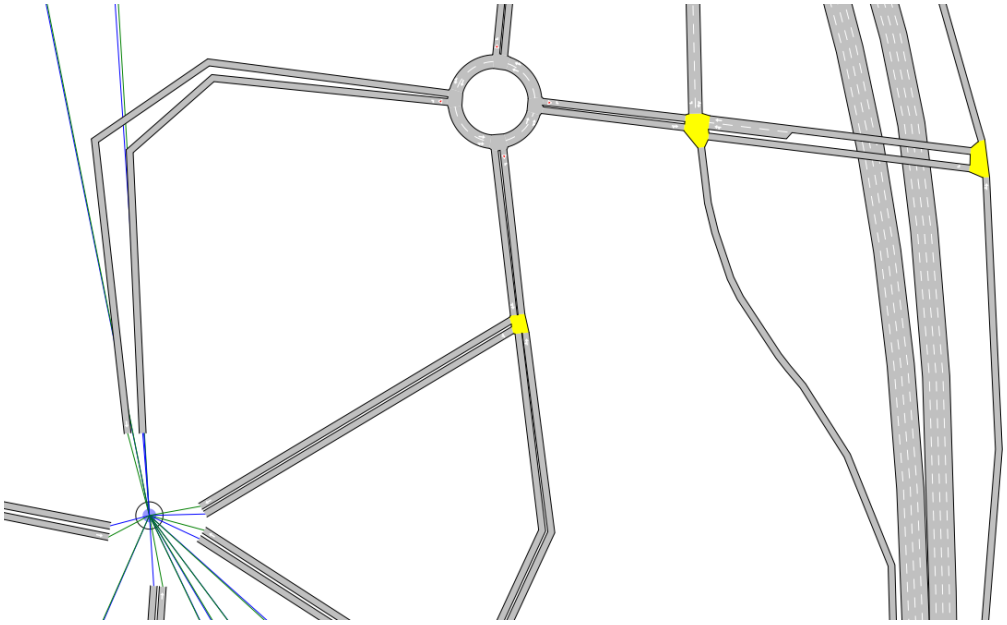

Figure 3.2: Detailed view on the modeled centroid in "Solna center"

number of sections. The percentages of how much traffic is guided to a specific section is stated by a table in the centroid's menu. In the project several sections are connected to the same centroid to keep the demand matrix simple. The detail of provided input data does not cover each origin in Solna. Thus it makes sense to simplify the model this way. Generating traffic with origin and destination in the same centroid is not possible. Short distance traffic is excluded from the simulation for that reason. Coming back to the model's purpose, the focus is on work based trips that typically do not stay in one neighborhood. In total there are 6 centroids included in the model. One is placed at the north and south end of E4, as well as the west and east end of the crossing E18. One more covers all Solna and the last one covers Frösundaleden, an arterial road south of the neighborhood. Figure A.1 in the appendix shows the whole network as it looks in the model. The demands between the centroids are given by an OD matrix that is estimated based on road sensor data from the highway. There are 28 sensors installed on the simulated stretch. Twelve of them are positioned on the north bound lanes and sixteen in southbound direction.

3.1.3 Demand data input

The simulation software Aimsun accepts two different kinds of input data for a traffic model. The first one is an OD matrix, stating how much traffic is originated and destinated in each centroid. The second option are traffic states on the links leading into the simulation area in combination with turning proportions for each intersection. For the project, OD matrices are more fitting, since the available sensor data does not cover all links and especially the turning proportions are not well visible from it. Further, the OD matrices can serve as samples for later comparison. The research question, whether it is possible to reproduce demands from CDR requires such a comparable structure.

The input for the used OD matrix in this simulation study is sensor data collected from the highway stretch. The Swedish highway authority "Trafikverket" uses these sensors to monitor traffic situations in real time and to adjust variable speed limits according to the demand. For research purposes, the collected data is stored in a data base that is accessible after requesting an account. The sensors are installed on each lane and count every passing vehicle. They do not differentiate between vehicle classes. The data set provided from these sensors contains measurements of flow in vehicles per hour once every minute. The values for different lanes on one highway direction are averaged. As representative days, the five Tuesdays between the 26th of April and the 24th of May 2016 are chosen. Tuesdays are commonly chosen to reflect typical working commute situations. It is insured that no public holiday was among these days. The three dimensional matrix resulting from the data extraction contains sensor counts for each minute in the morning rush hour on the regarded Tuesdays. The covered time is from 6:30 until 9:30am. Thus, the morning rush hour and the more calm hours around it are covered by the data. It is transformed using a Matlab script that can be found in the appendix. First, the script filters the data for those sensors that are of current interest. It then takes the counts from all Tuesdays for one specific daytime and averages them. Thereby, it eliminates both the highest and the lowest value to keep the result from being overly influenced by stochastic phenomena. The necessity for interpolation can be seen in figure 3.3. Mainly due to malfunctions, there are valleys and peaks in single sensor counts at many times. Those are eliminated before calculating the average. Finally, the script averages the counts within an interval of 15 minutes. In the model, one OD matrix is used to represent the demand during this time period. On the one hand, this reduces the number of different matrices to twelve for the three hours of simulation. On the other hand, it also ensures a proper representation of short term changes during the period as can be seen in figure 3.3.

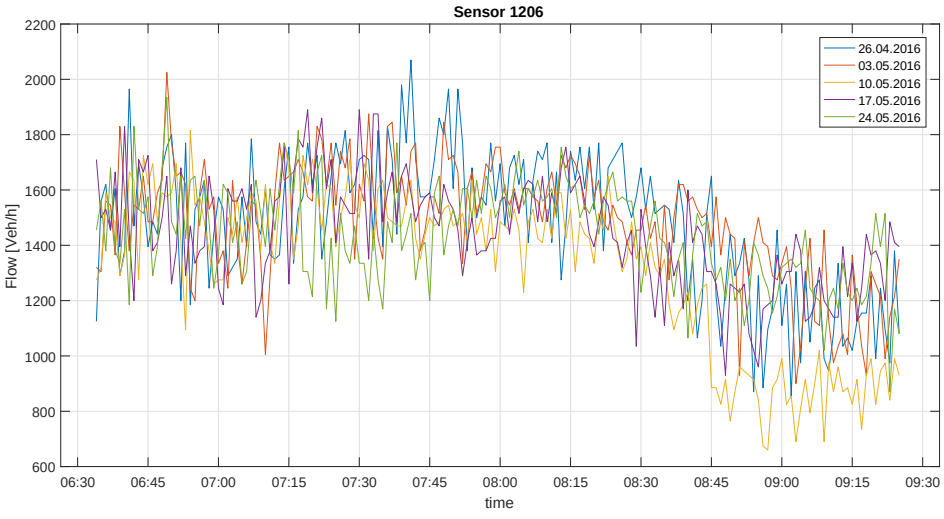

Figure 3.3: Comparison of data collected by one sensor on different days

Sensor data is only available for the highway and does not necessarily provide information about the OD pairs for single travelers. Especially for the residential area of

Solna, the data can only be derived from the differences between the highway counts before and after an exit and from experience about the typical traffic generation of this kind of neighborhood. Table 3.1 represents the initial demand at the beginning of the simulation at 6:30am. The OD pairs that include at least one of the ends of the highway can thus be derived from the sensor counts. The rest of the demands has been assumed. However, the demand trends from the Sensor data can still be applied to the rest of the traffic as well. The whole region is under the effect of the morning rush hour. Therefore, even the assumed demands can be scaled based on the fluctuation of the highway sensor counts. To do so, the resulting matrix is further processed with Excel as described in the following paragraph.

Table 3.1: Sample OD matrix [Veh/h], used from 6:30-6:45am

O/D	E18 west	Solna center	Frösundaleden	E4 south	E4 north	E18 east	Totals
E18 west	0	24	6	133	122	178	463
Solna center	25	0	21	51	51	25	173
Frösundaleden	9	22	0	18	18	9	76
E4 south	133	30	9	0	390	133	695
E4 north	141	26	8	390	0	141	706
E18 east	178	103	5	141	141	0	568
Totals	486	205	49	733	722	486	2681

To help understanding the temporary trends in demand, three indexes are introduced, each set to 100 for the first 15 min interval. They are displayed in figure 3.4. The indexes use the sensor data as input and translate the absolute differences of the highway flow into relative changes in traffic demand. Oscillations of the curves can be interpreted as relative changes in the flow. The peak of the "Northbound" curve at an index value of 123 for example represents a flow that is at 123% of the initial level. The first of the curves regards the overall traffic flow on the highway, measured by all sensors. The second one only includes those sensors in the northbound direction of the highway, as does the third one for the southbound direction. Since the flow is mainly generated by work commutes there is a significant difference regarding both directions as can be seen in figure 3.4. The indexes are used to adjust the twelve OD matrices. Demands that are assumed initially, get scaled by the index later. This way, the effects of the rush hour can be considered even for them. All OD pairs that are clearly directed in either the north or the south direction are scaled by the corresponding index. For the rest, the omni-directional index is used. All OD matrices can be found in the appendix. The peak demand is presented in table A.6. Between 8:00 and 8:15am, the time represented by the matrix, the highest demand is entering the simulation. The last OD matrix of the simulation covers the time from 9:15 to 9:30am. It is displayed in table A.11 and shows a decreased demand compared to the previous one. Finally the twelve calculated matrices are put into Aimsun as one combined traffic demand consisting of several scheduled matrices. The distribution of the demand within one period of 15 minutes is set to be uniform.

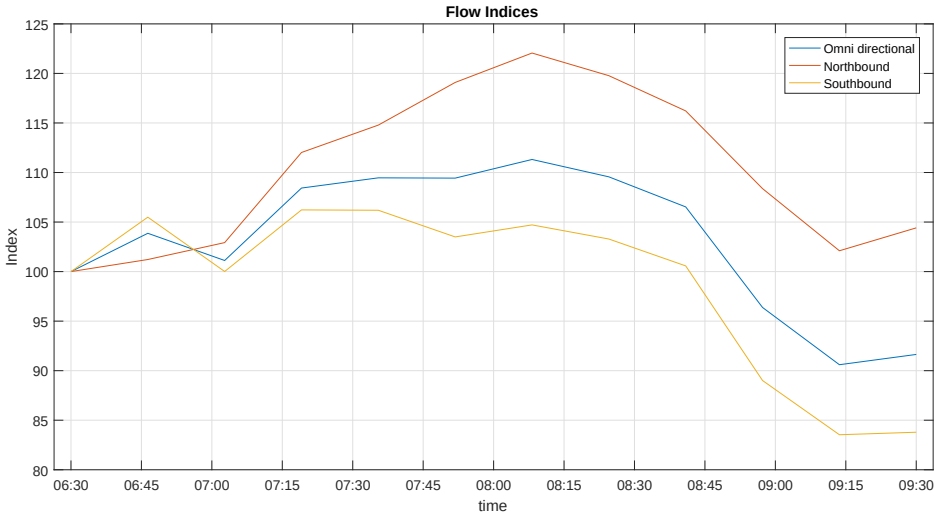

Figure 3.4: Development of highway flows during the simulation

3.1.4 Model parameter estimation

One of the most complicated procedures when implementing a traffic simulation is the correct estimation of parameters. Especially in a microscopic simulation, the correct reproduction of the original traffic can be complicated. The simulation is based on the reproduction of human behavior in traffic and this is hardly understood enough to ensure a proper initial choice of parameters. Hence, the validation of a simulation model usually ends up to be a sequence of trials and errors. Which parameters to manipulate in what way is mainly judged based on the experience of the engineer at work. Further it is crucial to have an understanding, which is the physical interpretation of the parameters that are used during the simulation. The Aimsun modeling handbook [63] gives an explanation what behavior each of them is supposed to represent. Furthermore, knowledge about the background models introduced in chapter 2.1.2 is helpful to estimate a model effectively. However, since even the best model can never be a precise representation of the reality, the purpose of the simulation study has to be kept in mind at this point. The estimation must be focused on the areas of main interest.

The microscopic simulation is, as introduced in chapter 1.3, the appropriate tool to generate a high number of vehicle trajectories. The specific application of the software requires a different approach to model estimation. It relies on direct observations during a running simulation. First, these observations are used to adjust some simulation parameters in several iterations. A Capacity of 2000 vehicles per hour on the highway is set to ensure realistic congestion conditions in the main peak hour and more free flow around it. Apart from that, the reaction time at stop is changed to 1.35 seconds. This helps solving unrealistic behavior of vehicles entering the highway. The simulated vehicles primarily use the rightmost lane of the highway and do not change lanes before an on-ramp to allow others to entry. Thus, the on-ramps tempt to get overly congested. The effects of this phenomena are lowered by the changed reaction time and a higher "Maximum

speed difference on ramp" of $70 km/h$ for the two lane car following model. Setting these parameters results in visible changes of traffic flow during the simulation. The procedure is very focused on the intention behind the study. The complicated and time consuming process of validating the model with given traffic counts is not performed. The estimation of the previously presented parameters is done on the base of observed changes of the traffic situation during interactive simulations.

In addition, a lot of attention has been paid on the route choice of travelers. By examining numerous vehicle trajectories through the animated simulation, it could be observed that many of them used the bypasses of highway exits and crossings as additional lanes to go straight. Normally these are only intended for exchanging leaving and arriving traffic. To avoid this behavior, the crossings had to be redesigned different from their original appearance. Further, it was observed that many vehicles took unnecessary detours to reach their destinations. As it is typical for urban highways with many exits on a short distance, the traffic guidance was quite complicated. Often the vehicles had to choose an appropriate lane long ahead in order to proceed to their turning. This was obviously too complex for the route choice model and resulted in the preference of a longer way using simpler highway exits. Hence, the corresponding exits were redesigned to fit better with the requirements of the route choice model.

The validation based on a real data set is skipped. The model includes urban arterial roads, residential streets and a highway. Only for the latter, traffic counts are available. Attempting to validate a model based on this does not necessarily increase its quality. Especially regarding the models purpose it is not helpful to spend time on the validation. For the intended research, the relations between local and highway traffic in terms of speed and traffic load should rather be correct. Furthermore, there should be different situations of free flow and congestion recognizable. To achieve this, mainly the demand proportions need to be represented correctly. By monitoring the traffic during several runs of the simulations, it can be confirmed that all cases are present in the study. An ongoing validation based on real traffic counts is complex for a study like this. Changing one parameter influences the traffic in such a diverse network in many ways. It can easily lead to unintended results and only a careful balance of many parameters leads to a real validation. Due to the limited resources available for a master thesis, the procedure is regarded as too time consuming. Furthermore, even with a validated model, a high number of replications would have to be to be simulated to get representative results. However, by running multiple replications, the actual trajectories of the simulated vehicles would be lost. They are different for each replication and thus only one can be used for the extraction. Stochastic influences prevent reliable results in one single simulation, even for a validated model. The whole output of the study depends on only one replication. For that reason, the traffic in the interactive view of the simulation is monitored closely, while running the study. It is ensured that the expected variety of traffic situations is present in the simulation run that the trajectories are extracted from.

3.2 The representation of the cellular network

Next to the simulation of traffic, there is another simulation of a cellular communication system included in the study. As the traffic model, it consists of infrastructure and a behavioral model. First, a representation of the system is imported to Aimsun. Second, a customized mobile connectivity model is developed. Based on this model, the cellular network is connected to the rest of the simulation.

3.2.1 Importing the cellular network overlay

Next to the classical parts of a microscopic simulation model, a representation of the cellular network is included into Aimsun. Including the network overlay directly in Aimsun simplifies the project significantly. Instead of first creating detailed trajectories for each simulation vehicle and exporting them to another program for generating the CDR data from them, both steps can be performed inside Aimsun. The level of detail in the final output is a lot lower than in the initial trajectories and thus a handover of large files can be avoided. Furthermore, the procedure is kept simpler and easier to reproduce by just using one software. However, before the cell towers can be imported into Aimsun as a layer, they must be prepared.

The source for the cellular specifications of the network is data from the website opencellid.org. This website offers location and specification data of mobile network cells that were collected by a community of users. The network providers consider their network infrastructure as a company secret and are not willing to publish the meta data of their antennas. However, every time a cellphone is connected to an antenna, the corresponding connection data is available on the device. Users who have an application of Open Cell ID installed automatically store and share meta data about their connection with the community. On the one hand this raw data can be downloaded from there. On the other hand, an algorithm combines all records related to one cell ID and estimates the locations of the cell's center along with its range from them. On an online map, the resulting positions of each cell are presented. Additionally, the processed database of cell locations is available as a download, too. The initial structure of the data set is presented in table 3.2.

The website only provides one database for the cell records all around the world. This results in a huge table that can only be handled by database tools like Microsoft Access. Filtering it for the relevant cells can easily be done by a simple SQL (Structured query language) statement. The data set includes the mobile country code (mcc) as a parameter. This way, it is possible to filter all cells from Sweden. Furthermore, the Location area code (LAC) stored in the *area* parameter helps to select all data from the region of Stockholm. To make a more individual selection of regional data, the data base can be filtered for a range of latitude and longitude values. Open Cell ID collects data for all network providers. However, for the project only one network is used. Connection records are collected from the providers and thus each of them provides a different data set in reality. Thus it is a reasonable approach to select one provider's network for the study. This network needs to provide a good coverage in the area and should preferably be used by a

Table 3.2: Open Cell ID data structure [51]

Parameter	Data type	Description
radio	string	Network type. One of the strings GSM, UMTS, LTE or CDMA.
mcc	integer	Mobile Country Code, for example 260 for Poland.
net	integer	Mobile Network Code (MNC) for GSM, UMTS and LTE networks. The System IDentification number (SID) For CDMA networks.
area	integer	Location Area Code (LAC) for GSM and UMTS networks. Tracking Area Code (TAC) for LTE networks. Network IDenfitication number (NID) for CDMA networks.
cell	integer	Cell ID (CID) for GSM and LTE networks. UTRAN Cell ID / LCID for UMTS networks, which is the concatenation of 2 or 4 bytes of Radio Network Controller (RNC) code and 4 bytes of Cell ID. Base station IDentifier number (BID) for CDMA networks.
unit	integer	Primary Scrambling Code (PSC) for UMTS networks. Physical Cell ID (PCI) for LTE networks. An empty value for GSM and CDMA networks.
lon	double	Longitude in degrees between -180.0 and 180.0 changeable=1: average of longitude values of all related measurements changeable=0: exact GPS position of the cell tower
lat	double	Latitude in degrees between -90.0 and 90.0 changeable=1: average of latitude values of all related measurements changeable=0: exact GPS position of the tower
range	integer	Estimate of cell range, in meters.
samples	integer	Total number of measurements assigned to the cell tower
changeable	integer	Defines if coordinates of the cell tower are exact or approximate. changeable=1: the GPS position of the cell tower has been calculated from all available measurements changeable=0: the GPS position of the cell tower is precise - no measurements have been used to calculate it.
created	integer	The first time when the cell tower was seen and added to the Open Cell ID database. A date in timestamp format: number of seconds since the UTC Unix Epoch of 1970-01-01T00:00:00Z
updated	integer	The last time when the cell tower was seen and update. A date in timestamp format: number of seconds since the UTC Unix Epoch of 1970-01-01T00:00:00Z
averageSignal	integer	Average signal strength from all assigned measurements for the cell. Either in dBm or as defined in TS 27.007 8.5 - both is accepted.

high number of users. That way, it is more likely that there is well funded data about the network available on the website. Therefore, the Telia GSM network has been chosen. It is identified by a value of one for the *net* and by the *radio* parameter. The result is a table with 1576 rows, small enough to open it with a GIS (Geographical Information System) Software or Aimsun. To evaluate the data it has been examined with the open source software QGIS. A map of the north of Stockholm, including the cells represented as dots can be seen in figure 3.5.

As expected, the density of cells is the highest in the city center. This represents a smaller average cell size in highly populated areas to increase the network capacity. Most dots are located on the big roads bypassing the center. This is mainly caused by the algorithm used by Open Cell ID to locate the cells centers. It takes all the measurements collected for one cell and locates the cell center in their middle without weighing them. Major roads with many travelers thus influence the position majorly and are more likely to be close to the estimated cell center. This phenomenom has recently been described by [49], who proposed an enhanced algorithm for future use. However, partly it makes sense to have a majority of cell centers on roads, since especially tri-sector antennas ori-

entate their cells along streets to avoid shadowing. It is important to understand that the dots in the maps do not represent the position of antennas, but of the estimated center of a cell. That way it is possible to have some of them in the water, without errors in the data.

The cellular network is imported as dots on a map into Aimsun. An example of its representation is shown in figure 3.5. The attributes of each cell are included as metadata automatically. While importing the location data to the program, it is possible to chose one parameter as "External ID". For the script to work properly, the *range* parameter should be selected as such.

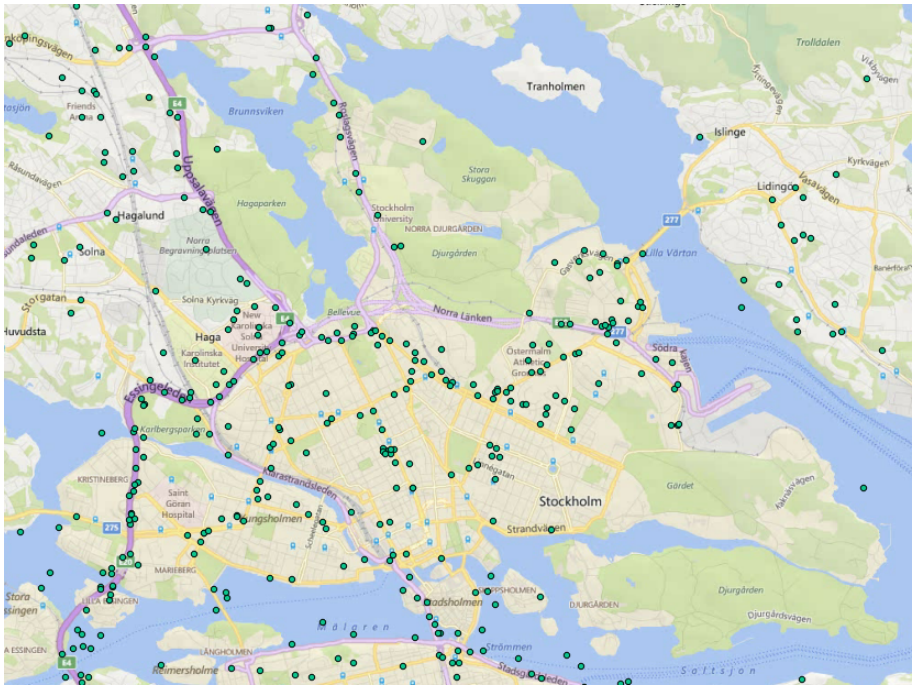

Figure 3.5: Map of Stockholm including the Telia GSM cells

3.2.2 Mobile connectivity model

The connection of the vehicles in the microscopic simulation and the cellular network overlay is handled by a mobile connectivity model especially designed for the project. The model follows the principles of network engineering that are introduced in chapter 2.2. At the same time, the requirements of the project are considered. The mobile connectivity model does not have to be applicable to real world networks. It merely has to mirror the behavior of cell choice algorithms given the preconditions of the project. The cellular overlay, as described in chapter 3.2.1 is a representation of a real network. The algorithms are especially designed to work with this representation. Additionally, the mobile connectivity model does not have to include the full functionality of its real world counterparts. Only a simulation of realistic decisions within the limited project environment is needed.

The baseline of the connectivity model is a cell choice following a voronoi logic. This logic is based on the distance between subscriber and cell center. A vehicle will, depend-

ing on its position, connect to the cell with the closest center. Thus, the research area is basically divided by discrete cell borders and the cells have a rather untypical polygon shape. A visualization of the voronoi polygon overlay in the research area is shown in figure 3.6. The map covers all cells that are relevant for the simulation area. Since the logic always establishes a connection with a nearby cell, the group of relevant cells can be easily determined. However, a strict voronoi logic neither provides realistic cell change behavior, nor does it consider the multi layer architecture of a typical mobile network. Therefore additional rules are applied.

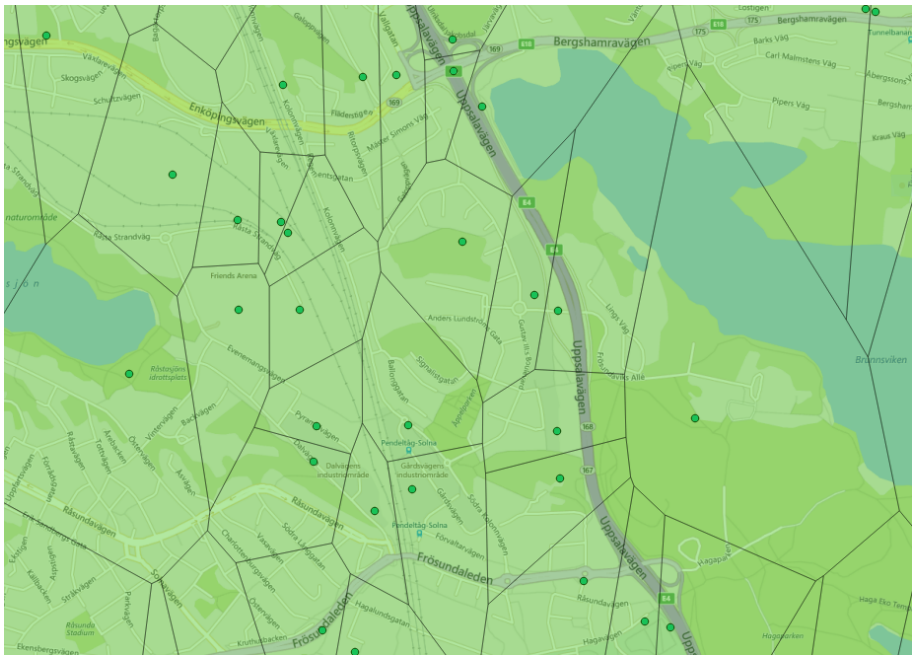

Figure 3.6: Voronoi cell structure in Solna

Two different cases are regarded in the model. The first one applies for all vehicles entering the simulation and connecting to a cell for the first time. In this case, the voronoi logic is followed. However, the group of feasible cells is filtered before based on the vehicle's speed. Fast moving vehicles are connected to bigger cells that offer coverage over a long distance. Slow travelers get connected to smaller cells that offer a higher capacity. This way, the multi layer architecture of cellular networks is taken into account. The second case includes those vehicles that already have been assigned to a cell and only change to another one when it becomes necessary. Especially in the case of cell change evaluation, a voronoi logic does not represent mobile network behavior well. An idea about how unrealistic the voronoi cell borders are, can be given by comparing figure 3.6 and figure 3.7. All over Stockholm there are multiple layers of cells on top of each other and not separated by discrete borders. A cell change will only be performed when poor signal reception makes it necessary and not as soon as there is a closer cell center available. Thus, limiting the relevant group of cells is a lot harder than in a voronoi logic. The cells presented in figure 3.7 are the subset of the Telia network that is used during the experimental procedure.

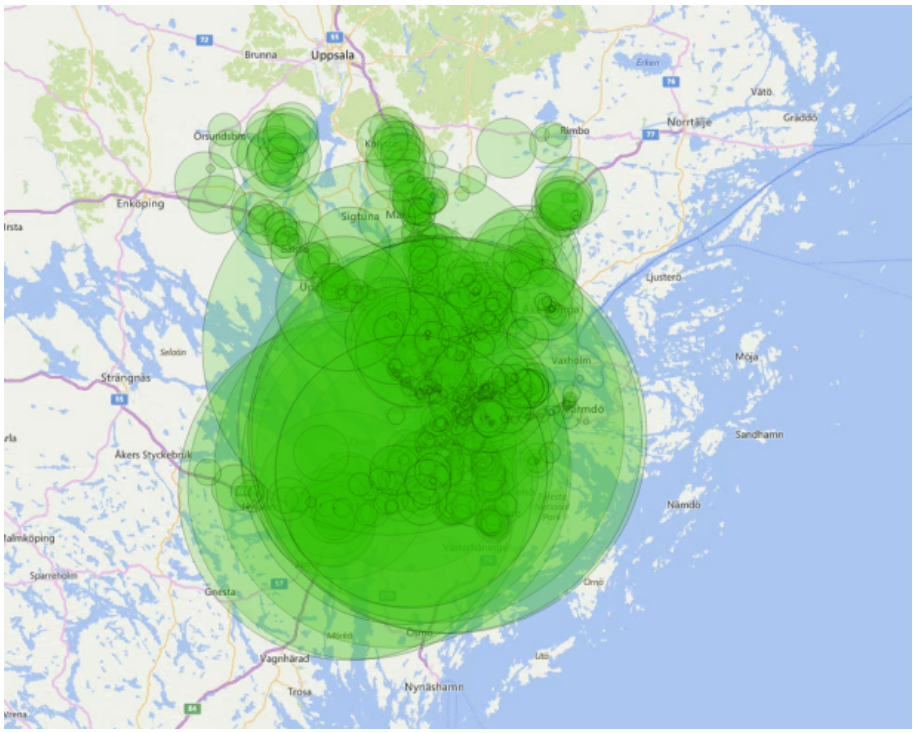

Figure 3.7: Multi-layer cell network in Stockholm

To represent the more reactive behavior that cell changes are based on, the model includes an additional first step. Each cell has a range attribute assigned to itself that represents its theoretical radius. As long as the distance between vehicle and cell center is smaller than the radius, a new call will automatically be handled by the previously subscribed cell. Only when the vehicle is outside the radius, a new cell is found based on the voronoi logic.

3.3 Generating the connection record

On top of the microscopic simulation model and the cellular network representation, a Python script is used to generate mobile connection records and export them from the study. This process includes several steps from the extraction of the vehicles' position, over the mobile connectivity model and the likelihood that a call is established. All of them are attended to in this section. Additionally, the general requirements of the Aimsun API regarding the structure of scripts are introduced.

3.3.1 Basics of the Aimsun advanced programming interface

The Aimsun API (Advanced programming interface) has been activated for the simulation and is an essential part in generating the CDR data set. The API makes it possible to

write extensions to the simulation in the programming languages Python and C++. The scripts contain predefined functions that are called by the software during the simulation. Further, Aimsun offers a library of predefined functions for both languages. These functions can be used to read and update all relevant simulation parameters. A list of all functions comes with each copy of the program and can be viewed and searched by using an internet browser. Since Aimsun is programmed in an object oriented way, there are classes and and sub-classes that inherit each others functions. Classes are the most basic form of of a definition for objects. All objects belonging to the same class will share the same attributes and functions. An example in Aimsun is the inheritance tree for a simple point, used for decoration in an Aimsun model. The point as an object belongs to the GKDPoint class for drawable points. It inherits from the classes for all graphical objects with a representation in 2D and/or 3D views (GKGeoObject). GKGeoObject inherits from GKObject that almost all Objects derive from. The Initial class is GKBaseObject that includes drawable and simulated objects. All functions for one of these classes can be used on any GKDPoint object as well.

A guide to get started with programming in Aimsun is provided in [62]. It includes all possible ways of coding in Aimsun and thus does not go into any details. It merely focusses on introducing the necessary tools and the programs internal data structure. The latter has to be known to call variables in a code. A more in detail introduction to the Aimsun micro-simulator API that is used in the current project, is given in the conferring manual [63]. It includes a wide range of functions and explanations to perform the most common and basic operations during a running microscopic experiment.

The basic structure of every script for the API is sketched in figure 3.8. The left half of the figure shows the steps during a simulation. The right half contains the functions contained in an API script and at which point in time they are automatically accessed by the program. Hence, this set of functions has to be part of every API and all code that should be executed during the simulation must be written inside one of them. The default return value of every function is *0* which has to be kept in order to acknowledge that the function has been executed. In the preamble, the AAPI library must be imported. It contains all Aimsun specific functions and objects.

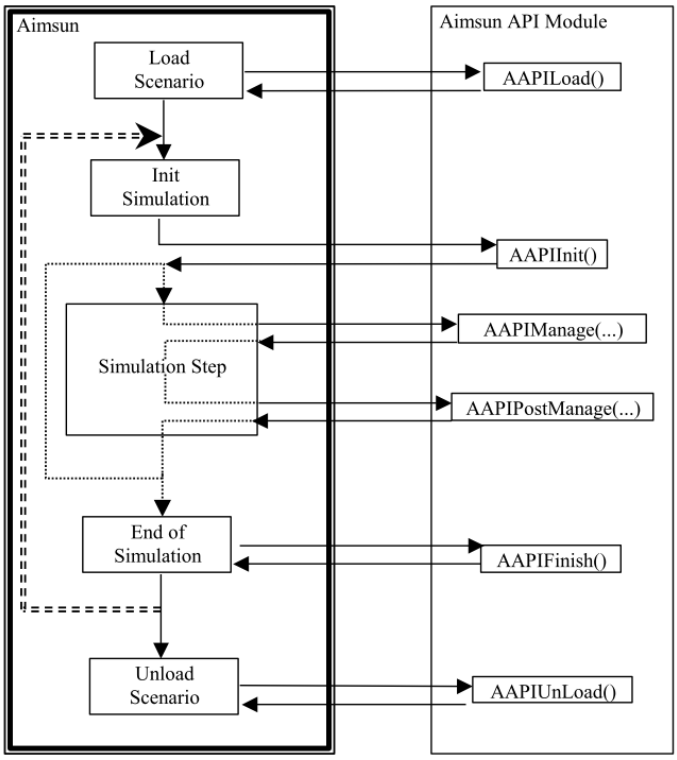

Figure 3.8: Communication between Aimsun and the API during a simulation [63]

3.3.2 Call likelihood model

The complete script used in the project is attached in the Appendix. As a programming language, Python has been used. The main purpose of the script is to translate data into a specific format. In many places this means to abstract the data in a certain way. In some points the user can influence the degree of data scrambling, in others it happens automatically due to program requirements.

In the Preamble, the *datetime* library is imported in addition to the default. Its methods are needed to add the time stamps in the output file. The rest of the code that is placed on top of the functions is used to declare variables. This is done to make them globally available during the whole simulation. Basically all variables in the script can be set by the user to balance the level of output detail with the required processing complexity. A trade off that is critical, since Aimsun tempts to crash when the computers capacity is exceeded. For that reason, *Cycle* determines the time gap between two data extractions. This means that positions are not extracted at every simulation step. *Cycle* stands in direct correlation to *relative*. This variable determines the relative amount of users establishing connection during one time step. *Relative* should typically be lower for a smaller *cycle* time. Since the script is used to gain experience in working with different CDR data sets, the setting of both variables is left to the user. In this project a cycle time of 17 seconds and a relative call likelihood of 0.2 are applied. These values are aimed to represent real driver behavior and to generate an average number of 2 connections per vehicle in the simulated network. Experiments with different values are not the focus of

this research project. The two variables *SG* and *PG* refer to groupings created in the Aimsun simulation. *SG* is a grouping of sections and specifies the research area. It enables to pick only a sub area of the whole model for data extraction. This way, one model can be used for several different experiments. *PG* defines a sub set of cell towers to use in the experiment. Reducing the number of towers increases the scripts performance. Furthermore, experiments with different network providers in the same model become possible. The values stored inside *SG* and *PG* are the groupings' automatically assigned IDs.

In addition, an empty .csv file is created to store the results in. Within the script it can be accessed by the variable *filename*. To refer to the active model from the script, it is assigned to the variable *model*. The model with all its parameters is stored as a GKModel object. Hence, *model* is more than just a single value and several functions can be used on it.

Within the first function *AAPILoad()* that starts in line 17 and is called in the very beginning, the headline of the output file is generated. Therefore, python opens the file in write mode. CDR data typically consists of 3 columns: User ID, cell ID and time stamp. In the output file a fourth column with the previously connected cell ID is added for diagnostic purposes. In order to save the changes made to the file, python requires to close it after use. It is crucial to close all used files before finishing the script. Thus a command in this regard should be added in the end of each function that accesses one.

The rest of the program is written inside the *AAPIPostmanage* function starting in line 29. Hence, it will be executed repeatedly, directly after every step of the simulation. A process flow diagram of the following steps taken within the call likelihood model can be found in figure 3.9. All global Variables are made available for reading and writing. Next to them, some automatically by Aimsun generated variables are handed over to the function by default. *time* is the current duration of the simulation in seconds starting from 0 whereas *timeSta* transforms the simulation daytime into seconds. *timeTrans* and *acycle* are parameters stating the duration of the warm up period and of one simulation step respectively. As first action, an *if* statement is executed in line 31. It affects the whole following code inside *AAPIPostmanage*. The statement puts the *Cycle* variable to use and checks whether it is time to extract data or not. If yes, a time stamp for all outputs is generated and the output file is opened again in line 33.

The second *if* statement in line 42 applies the overall likelihood of a connection for each user. This statement is positioned inside a loop through all vehicles, since it is decided for every single one of them. A random number between 0 and 1 is generated for the vehicle in line 41 using the *AKIGetRandomNumber* function of Aimsun. Only if this number is smaller than the *relative* variable defined by the user before, the record is generated. Otherwise, the *break* statement in line 43 skips all the rest of the indented code and continues with the next vehicle. Unaffected by this term is of course the closing of the file in line 77.

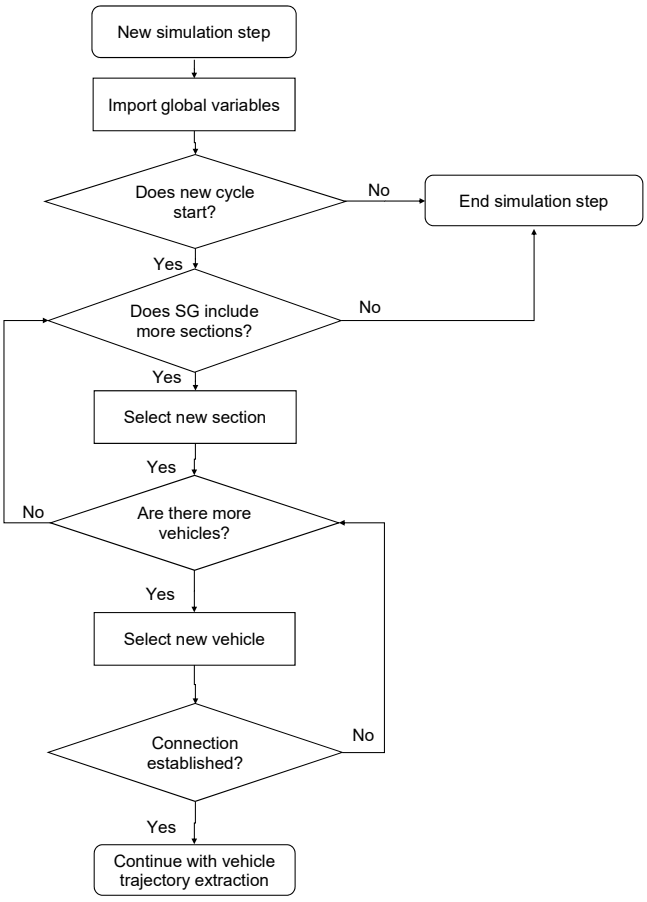

Figure 3.9: Process flow diagram of the call likelihood model

3.3.3 Vehicle trajectory extraction

Reading the location data of each vehicle is the first step towards generating the mobile phone connection record. By saving detailed trajectories first, several experiments can be run on the same data. Those trajectories are then to be transformed using a mobile connectivity model in a second independent step. This procedure keeps the data to compare the record with the actual path and to find connections between both. An example how to export the trajectories is given in [61]. The script presented there, utilizes the simulated vehicles to get a position in geo coordinates. In an Aimsun microscopic simulation each vehicle is represented by two identities. One is the identity inside the model. It contains all parameters and the background models are applied to it. The other identity is the vehicle's optical representation that can be seen on the screen during a simulation. This identity consumes a lot of processing power and is only generated, when an interactive simulation is performed or if it is specifically requested by a script. The vehicle positions in geo coordinates are only available from the simulation vehicle. With those geo coordinates the distance between vehicle and cell tower can be calculated more precisely.

Although it would therefore be preferable to use the simulated vehicles, it leads to instability of the program and can hence not be done. Without utilizing the simulated vehicles it is possible to get the positions relative to the borders of the research area. These are not a valid reference for other programs, but can still be used inside Aimsun. To do so, first the vehicles of interest have to be selected.

The process of extracting the vehicle trajectories from the simulation requires a lot of transition of data types in the script. This chapter explains the procedure in detail to enable the reader to understand all steps taken in the script. In terms of logical steps, the trajectory extraction simply takes those vehicles as input that were determined by the proceeding call likelihood model. It then browses the models catalog for them and looks up their positions inside the research area. This position is used in the following mobile connectivity model to motivate the decision for which cell tower to connect to. The following paragraphs point out how this is achieved in the script. Initially the *sections* variable is declared in line 34. Therefore, the *GetCatalog*function is called on *model*. The catalog contains all parts of a model, like for example the groupings that have been created before. By using *find(SG)* on it, the corresponding section grouping can be found. The grouping is not identically to its ID stored in *SG*, but is rather a shell for all its containing objects. Finally, the *GetObjects* functions opens this shell and returns the sections as a list. In Python it is possible to loop through a list. The loop through *sections* commences in line 37. From there on the currently regarded section is represented by *i*. Calling the function *getId* on it returns the sections ID as a number and stores it in *id*. This is needed as input for the next function *AKIVehStateGetNbVehiclesSection*. It finds the section in the current simulation's data base and returns the number of *vehicles* currently present in it.

This number provides the input for the next *for* loop. Since such loops only work with lists, the *range* function is called to turn *vehicles* into a list of numbers from 1 until *vehicles*. After the following *if* statement that is part of the call likelihood model, the current vehicle is looked up using the section *id* and its number *j* inside this section. The function *AKIVehStateGetVehicleInfSection* is included in the Aimsun library for that purpose. The result *probe* is saved as a vehicle object and thus can have functions called on it. *probe.idVeh* that is called in line 45, returns its ID. *probe.xCurrentPos* and *probe.yCurrentPos* return the vehicle's current position. Since *probe* is not a simulation vehicle, the position is expressed in a 2 dimensional coordinate system with the origin in the bottom-left corner of the research area as it is displayed on the computer screen. Based on the x and y position, Aimsun can create a *GKPoint* object and save it as a reference position. Since this procedure is repeated for every simulation step, the sum of points creates a trajectory. The loops through all sections and vehicles ensure a high number of trajectories. However, since the positioning data cannot be displayed on a map, it is useless outside the program. Thus, it is further processed by using the mobile connectivity model, before it is saved.

3.3.4 Implementation of the mobile connectivity model

The mobile connectivity model is applied on the vehicle's location starting in line 47 of the Python code found in the appendix. Before, an empty dictionary *Connectionrecord* is created. A dictionary assigns values to keys. The values can be looked up by providing the corresponding key. It is used in the model to store the previously assigned cell for each vehicle. Further the variable *PG* represents the ID of the grouping of active cells in the current simulation. A process flow diagram of the mobile connectivity model can be found in figure 3.10. The left half of the diagram represents the connection to a new cell, while the right side covers the continuous connection to the previous one. The diagram is intended to help understanding the logical steps of the program, while its details are explained in the following paragraphs. All references to the script concern the Python script for CDR generation that can be found in the appendix. While the theory behind the model is described in chapter 3.2.2, its implementation in the script is described in the following paragraphs.

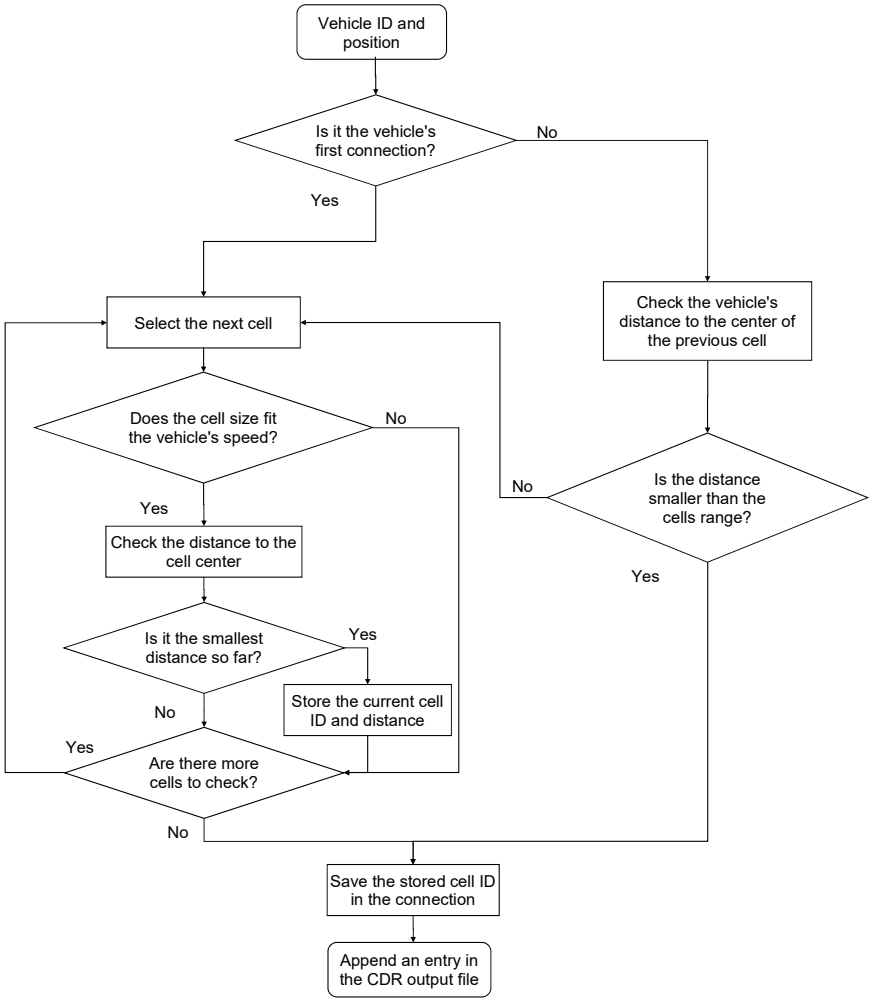

Figure 3.10: Process flow diagram of the mobile connectivity model

The mobile connectivity model regards two cases. The first concerns new subscribers that establish a connection for the first time. The second case regards those that were already connected to a cell while being in the geographic research area. The decision, which off them applies is made by the *if* statement in line 48. It distinguishes whether the ID stored in *probeId* already has an entry in the dictionary *Connectionrecord*. Since the ID of modeled vehicles is unique, it can be used to make this decision. The code is wrapped by two loops and thus executed for every vehicle in every section that is part of *SG*.

Within the first case, another loop iterates through the list *points*. *points* has previously been generated in line 35 by extracting all objects from the grouping identified by *PG*. Thus it represent the collection of all active cells. The loop is used to find the most appropriate cell to establish a connection with. First the *radius* of the currently regarded cell *k* is extracted. It has been assigned as the points external Identifier when importing the points into Aimsun. Hence, it can be called by the function *getExternalId*. The following *if* statement is used to apply the multi-layer network theory, presented in chapter 2.2.3. It does so, by comparing the size of a cell with a vehicle's current *speed* in km/h. A cell is only applicable if the following expression is true:

$$0.7 < (speed^2 + 500)/radius < 1.3 \tag{3.1}$$

This formula has been scaled to include all available cell ranges in the typical range of speeds between 0 and $120 km/h$. Figure 3.11 visualizes the selectable range of cell sizes for each speed. It also includes a plot of the sorted cell sizes that are available in the cellular network. This plot is scaled to the same x-axis. The formula is developed specially for the present mobile connectivity model. Powering the *speed* by 2 in the calculation ensures that the biggest cells are not prioritized by the choice algorithm. Generally adding 500 to the result ensures that slower travelers still find valid cells to connect to. As figure 3.11 shows, the feasible cell sizes represent the distribution of radii in the cellular network. The tolerance for the result between 0.7 and 1.3 represents a trade-off between avoiding unnecessary handovers and effective cell filtering. The boundaries are set to reproduce realistic Cell dwell times of 3 to 7 minutes when traveling at constant speed. This way traveling users can have a call, without experiences many handovers while it lasts. Even though the handovers are not part of the model itself, a realistic cell choice has to take some handover balancing into account. At the same time the boundaries ensure that every vehicle within the simulation is able to find a feasible cell to connect to. The parameters of the equation are scaled regarding those two objectives. All valid cells are hereafter compared by the voronoi approach. Line 54 assigns *poinPos* the cell center's position as a GKPoint object. Since *pointPos* and *probePos* are both available in this format, Aimsun's *distance2D* function can be used to determine their two dimensional distance in meters. It is checked whether this distance is smaller than to any other valid cell center so far. If yes, *k* is saved as *Celltower*. Since the procedure is repeated for all cells, *Celltower* will be updated until the best candidate is found. Once the loop has ended, the vehicle's ID, the time stamp and the Name of the closest cell are stored in the *Results* file. This represents the creation of one entry in the CDR data set. At last, an entry in the *Connectionrecord* dictionary is created, assigning the key *probeId* to *Celltower*. Ergo a record will be found next time and the second case of the mobile connectivity model executed.

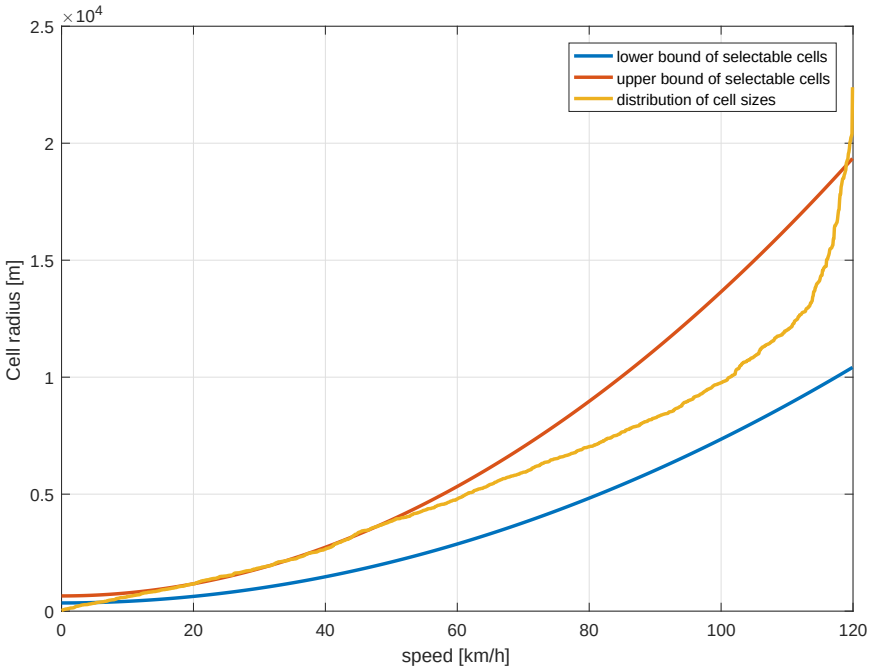

Figure 3.11: Development of the cell choice formula

Starting from line 61, it is checked whether the vehicle is still within the radius of the previous cell. If so, a cell change is prevented by setting the distance to the cell center to 0. That way the following voronoi logic cannot find a better solution. At this point the model profits from its Open Cell ID input data. There, the cell radius is estimated by the furthest distance of any mobile unit that was reported to be connected to the cell. Hence, this value ensures that a connection is realistic within the radius. In case the previous cell is out of range, the same procedure as in the first case is applied to find a new one. The two presented cases are the only ones relevant regarding a generation of CDR. Handovers during an active call do not need to be regardedby the model, since only the first cell of each connection is stored in the data set.

Chapter 4

Results

As result of the experiment, synthetic CDR data sets are generated. This chapter focuses on the data itself and evaluates its characteristics. A first part describes the data's structure and lists all the files that are available. Knowing, what is the shape of the input, several approaches are conducted in the following sections. Each of them runs a different analysis and evaluates its results. All investigations are related to traffic analysis and how CDR data can be used for it. Since the current data sets are only synthetic samples that cannot represent reality, the analyses are performed in a way that keeps possibly more randomized input in mind.

4.1 Structure of the output data

In this chapter the files are presented that are at the same time output of the experiment and basis for the following analysis. The files are generated using the tools and techniques presented in chapter 3. The simulation is performed in interactive mode, including simulated vehicles, in order to more realistically capture the actual traffic situation. In total, the study is run three times using different OD matrices. First, the set of matrices as it was estimated based on the road sensor data in chapter 3.1.3 is used. This scenario is hereafter addressed as the *Original Scenario*. Additionally, the *Free flow Scenario* applies demand matrices that share the same weighting between the OD pairs as the original one. The total demands however, are lowered by 20%. As the scenario's name suggests, this results in free flow conditions for almost all parts of the simulation area. The third scenario on the contrary deals with a 20% increased demand throughout the whole simulation. In this *Congestion scenario* all major roads face seriously increased travel times in the later parts of the study. An example for the traffic situation in the simulation study is given by figure 4.1. It displays the average flow per hour on every link during the simulation period. The coloring marks how the flow is related to the sections' capacities. As expected from the input data, the highest numbers are reached on the southbound highway.

Within each of these scenarios, four scripts run simultaneously. They all generate CDR data as presented in chapter 3.3. The difference between them is the group of sections that is regarded. The first script includes all sections within the whole research area.

Figure 4.1: Capture of the network summary from the *Original Scenario*

It provides the biggest number of samples for analysis and is the closest to a real data set as possible in this project. The other scripts only generate data from parts of the network. One of them includes all residential streets inside Solna. They are all characterized by a low capacity and slowly moving traffic with a big diversity in movement direction. Another script on the opposite regards all sections of the highway and its ramps. The traffic included in this script is typically moving at high speed and is limited to the north and south direction. An even more focused view on the highway gives the fourth data set that only includes the highway itself. Due to the unified traffic and its narrow specification, this data set is good to observe highway specific connection characteristics.

All generated tables are stored automatically as *.csv* files by Aimsun. A sample of one of the files can be seen in table 4.1. It is filtered to show the records of five randomly selected vehicles. Each of them has created either two or three entries. The column *Vehicle ID* represents the anonymous caller ID. The*time* column provides the time stamp with a precision of one second. *Cell ID* is the identifier for the currently connected cell. The column *origin* is not included in real CDR data sets. It is added here to visualize the cell change process. Whenever the value for *origin* is Error, the vehicle just entered the simulation and has not been connected to any cell before. The size of the data sets varies between 2000 and 15000 entries as can be seen in table 4.2. Typically, the sets from the *Congestion scenario* contain most entries compared to their counterparts. While the difference is significant for most data sets, it is only marginal in the case of the *highway* table. However, the sizes give a first idea, how different densities affect a CDR data set and why it can be useful for traffic planners.

Every further analysis of the results is done with the help of Matlab. Thus, the files are imported into the software using the Matlab data importer. For the software the files contain different data types, as *datetime* for the time stamp and *String* for the origin column. It therefore can only be imported as a table. The headlines, found in the first column serve as variable names. Tables only allow very few operations and therefore have to be transferred into column arrays for any kind of analysis. However it is helpful to have each data set stored as one file in the beginning.

Table 4.1: Excerpt from the *Congestion everything* CDR data set

Vehicle ID	time	origin	Cell ID
2112	06:41:56	Error	23052
2112	06:43:21	23052	13232
5731	07:08:00	Error	13200
6755	07:09:25	Error	13200
6755	07:09:59	13200	13160
5731	07:10:50	13200	44961
6755	07:12:32	13160	13160
18332	07:41:26	Error	13200
18332	07:45:41	13200	44961
34410	09:06:26	Error	45700
34410	09:06:43	45700	43093

Table 4.2: Number of entries in each generated data set

		Scenario		
		Original	Free Flow	Congestion
Data Set	Everything	11973	9019	14392
	Highway	2204	2172	2201
	H. and Ramps	5554	4769	5627
	Residential	3653	2234	4471

4.2 Output data analysis

In order to discover relations between the CDR and the traffic situation it is based on, the data sets are evaluated from different perspectives. Each of the following sections regards one analysis approach and evaluates its meaningfulness. The comparison of relevant data sets from all scenarios give an estimate of the results' sensitivities towards changing traffic conditions. The analysis is carried out using Matlab scripts that are designed to work for any kind of data set generated with the method presented in chapter 3. Thus they can be conveniently reused for ongoing research.

4.2.1 Total system load

From a mobile networking perspective, the research area can be described as a set of cell towers that are used by the travelers during their commute. The multi layer design of cellular systems is introduced in chapter 2.2.3. This paradigm results in cells that are primary used by traveling subscribers. It is assumed that the subset of these cells is directly affected by a changing traffic load and hence can give an estimate of it. Every vehicle has a relative chance of receiving a call as long as it is inside the simulation. Thus, the total number of records in the system gives an estimate of how many vehicles are located inside the area. It is therewith related to the density of the traffic on the sections. The more records are produced, the higher is the density and vice versa. This theory is put to a test by plotting the total load of the cells used during the simulation. Since the cell choice algorithm is well known for this project, the results could easily be improved by just applying the same algorithm again. However, that would undermine the meaningfulness of the results. Thus, the analysis is limited to what can be directly observed from the data sets.

Figure 4.2 presents the results of the analysis. It compares two graphs. The first one regards the data sets *Everything* and the second one specifically the *highway and ramps*. As for the first graph, a clear difference between the curves can be seen. As expected, the number of records and thus the density the biggest in the *Congestion* scenario, followed by the *Original* and the *Free flow* one. It can even be observed, how the curves spread more as time passes and congestion builds up differently. However, while only regarding the highway and its ramps, the picture is not as clear. A separation of the curves like in the first plot cannot be observed here. Since we see that some connection between density and number of connections exists the reason must be that the jam density on the highway is not significantly bigger than during dense, but flowing traffic. This theory is supported by the previously discovered effect, that the highway data sets have almost the same number of records in all scenarios. From the simulation it can be observed that the bottlenecks are the intersections between on ramps and the highway. Before those spots, the traffic gets jammed first. However, many spots on the highway remain deserted during these times. Thus, the overall density on the simulated highway stretch is probably not bigger during congestion. This phenomena cannot be taken as a general rule though, it may as well just be related to the limited area of the model or the behavior of the simulated vehicles. In how far the number of connections can help identifying different traffic situations, depends on the shape of road at hand and requires experience with its bottlenecks.

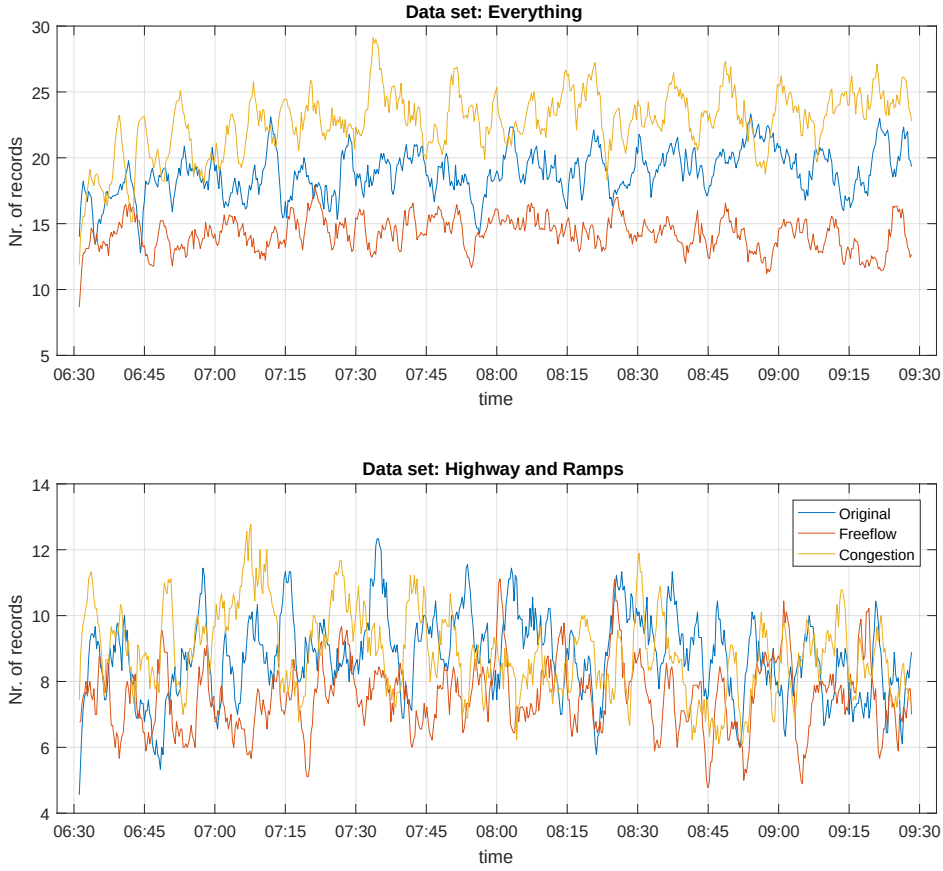

Figure 4.2: Comparison of plots for the total number of connections

4.2.2 Average cell size

Cell sizes differ in a wide range and each kind of cell serves a specific purpose inside a network. The cell sizes are basically a trade of between capacity and coverage. The coverage of a large area can be important to avoid a high number of handovers for traveling users. Thus, handover algorithms are commonly programmed to connect those users to big cells, while stationary ones are assigned to smaller cells with a focus on capacity. The ways, how handover algorithms detect the movement differ widely, but the strategy behind them is the same. Following this model, the assumption for this analysis is that the radius of the currently connected Cell is dependent a vehicle's speed due to the choice pattern of the handover algorithm. As long as most traveling users are connected to large cells, the average speed of vehicles is expected to be high. As soon as the algorithm starts to assign smaller cells, this means that the vehicles' average speed has dropped. This way, the average size of cells that are used by vehicles in the simulation at every time can contain information about the traffic conditions. To support the thesis, several data sets are compared regarding the average size of their providing cells.

An overview of this comparison is presented in figure 4.3. The graph visualizes the

average cell radius of connections from the *Free Flow Scenario*. Due to the constantly non congested conditions there, a good sample for the typical radius bands for each road type can be observed. The plots for the different data sets are clearly separated and all remain leveled throughout the whole study. It can be observed that cell sizes above 9000 are primary used by vehicles traveling on the highway. It also appears that the average cell size even for urban traffic does not drop under 2000. The third curve for highways and ramps shows that even a less homogeneous data set fits in the picture. It includes the fast highway traffic and slower one from the ramps. The average of this stays somewhere below the pure highway data.

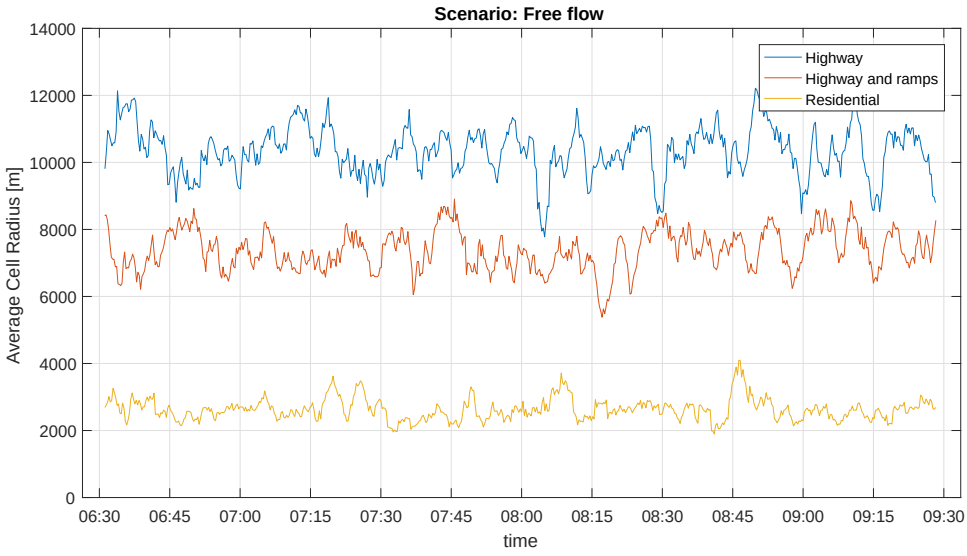

Figure 4.3: Comparison for the average size of serving cells in the *Free flow Scenario*

The graph in figure 4.4 is used to observe the effect of increasing traffic on the average size of active cells. It compares the *Highway* data set of all three scenarios. While all curves begin at a radius around 10000, the *Congestion* scenario's quickly decreases to values around 6000. The *Original* curve shows a similar, but less strong reaction and levels around 8000. This curve even raises again in the end, a behavior that fits quite well with the expected amount of congestion. Since the simulation period regards the morning peak hours, the traffic load is expected to decrease at its end.

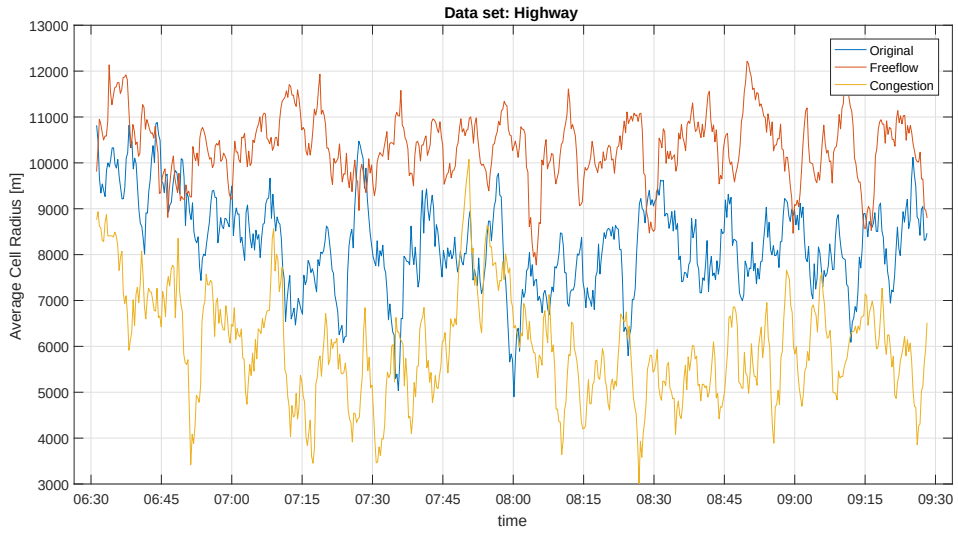

Figure 4.4: Comparison of the average size of serving cells for highway traffic

Finding these patterns in a more randomized data set that does not only include highway traffic is hard. When the travelers on a highway only count for a small part of the entries, it should be impossible to identify congestion only based on the change of the curves. However, this analysis can be very useful to filter the highway data out from all the rest. Since there is no mode that allows a comparable high number of people to travel at high speed, all the biggest cells should serve mainly highway travelers. Congestion can hence be identified by monitoring the biggest cells. They are only used by fast highway travelers and the number of those massively decreases when the highway is congested. The slowed down vehicles will hence rather subscribe to smaller cells. The more idle the biggest cells hence become, the more congested the highway is. For this method to work, a multi-layer cell network is required. Therefore it is mainly applicable for urban highway stretches.

4.2.3 Cell dwell time

The time, a subscriber is connected to one cell is an important measure to evaluate handover algorithms. For the purpose of traffic planning it can be helpful regarding the traffic conditions on a road network. The faster a subscriber is traveling, the sooner he will be out of one cell's range. Hence, the cell dwell time becomes shorter. The multi layer network in the geographical research area is not helpful regarding this analysis. since faster cars are connected to bigger cells and their dwell times are increased by that. However, a relation is still likely. Especially traffic jams with almost no progress should be possible to identify.

A similar approach to conduct this analysis is to simply count the number of connections per car as long as it is in the simulation. It is less transferable to a real scenario, than the dwell time approach, because a limited geopgraphical area is needed to get compa-

rable numbers for total counts. The sum of connections is only used as a comparative value to the cell dwell time. What becomes apparent while running the analyzes, is that the likelihood of an established connection is quite small to deliver meaningful results in most cases. In the Free flow and the Original Scenario, cars often don't spend enough time inside one cell to generate more than one record. This is crucial though, to calculate the dwell time. For the scale of this project, the analysis is hence focused on the Congestion scenario and the data set including all links. By regarding all links, the time inside the area is maximized. Since the average speed in the congestion scenario is the lowest, also the time in one cell is longer. A plot comparison for the average dwell time and the number of records per car for this scenario can be found in figure 4.5. The blue curve shows the average number of records of each car for every minute during the simulation study. The red curve shows the cell dwell time respectively. It is clearly visible how both, the connection count and the dwell time rise significantly as soon as the peak hour begins. The data is jumpy, but still the difference between the un-congested beginning and end period and the peak hour is evident. The sensitivity of the cell dwell time to traffic changes appears to be higher than of the connection counts. Both values are listed at the time of the first occurrence, this can hence not be the reason for the typically earlier oscillation of the dwell time curve.

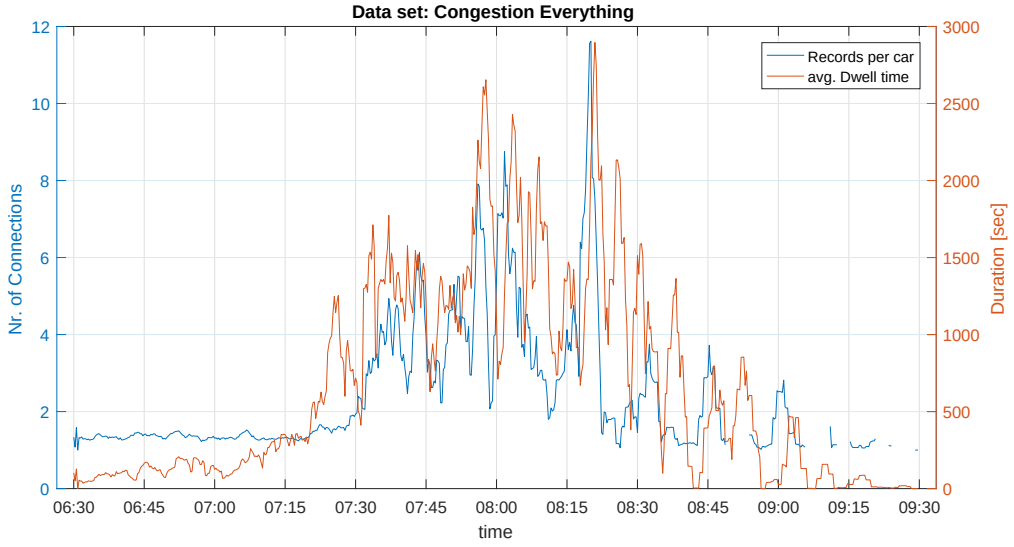

Figure 4.5: Comparison of dwell time and number of records for one data set

The part of the plot up to minute 40 in figure 4.5 gives a hint about how the other data sets appear when being analyzed. The average number of connections for both the *Free flow* and the *Original* scenario remain at 1.5 the whole time. The average cell dwell time of these scenarios lies at 74 and 98 seconds respectively. A slight difference between the two can hence be observed. For comparison, the average number of records in the *Congestion* scenario is 2.2 and the average cell dwell time 672 seconds. Again, the difference is more obvious by the dwell time curve. This makes it a very promising approach to follow for bigger data sets. Especially in rural areas, where the cell sizes are bigger, it can deliver meaningful results.

4.2.4 OD estimation

One of the most common applications of CDR data in traffic models so far is to generate OD matrices from it. Through long term monitoring of the users, their daily commuting behavior can be observed. An overview of the most common approaches can be found in chapter 2.3.2. The data sets in other studies focusing on OD matrix generation usually include much bigger areas. This lowers the drawbacks resulting from the imprecise localization. However, even from a small data set, it might be possible to conduct some travel directions. The rather simple shape of the road network in the simulation study is an advantage for this analysis. The highway runs along the north-south axis and the other major road E18 along the east-west one. Therefore it is assumed that all traffic that clearly goes in one of these directions is based on those roads. In the simulation, there are centroids placed at all four ends and the demand data hence is known and can be seen in the referring cells of table 3.1. The experiment is limited to only these two OD pairs, since the rest of them does not infer a simplified travel direction. The more centroids are included in such an experiment, the more complicated it becomes to address the records to the right OD pair. Therefore, a first attempt shall focus on the presumably most easy situation of traffic on E4 and E18. In case this succeeds, further OD pairs can be included in the experiment.

The localization of the records is based on its related cell ID. For each ID, a location is stored in the Open Cell ID data base. Following the path of a vehicle thus means to follow the connected dots of cell centers. For the OD generation the first and the last entry of each vehicle is utilized. The longitude and latitude of both connections are subtracted from each other. This is possible only if the vehicle is connected to at least two different cells during the simulation. This is not the case for most vehicles though. It means that a driver has to establish at least two connections while being in the simulation area and that the phone has to change cells in between. Since the overall likelihood is low, a big data set is preferable for the analysis. The more roads that are included in the data set, the longer an average vehicle will stay inside its range. Furthermore, a high number of samples increases the chance of repeating unlikely cases. The cell tower locations are provided in terms of longitude and latitude. The distances between the towers hence has to be converted. The longitudinal meridians move closer together while approaching the earth's poles. Since Stockholm is positioned at 59° north, the effect is considerable. The distance between two meridians is a lot smaller here, than it would be on the equator level. The scaling factor to convert longitude difference to meters here is 0.0435, while it is 0.1115 for latitudes [68]. The resulting distances are compared and the bigger absolute of them determined as the vehicles movement axis. Whether the distance is positive or negative decides its boundary direction.

Table 4.3 shows the results of an OD estimation based on the *Original Scenario* including all links. The column *CDR* includes the total count of vehicles traveling in each direction. As a comparison, the *Sensor* column provides the related values from the original OD matrix based on sensor data. Each value pair has a theoretical scaling factor that

would be required to scale the CDR count to the sensor values. Some scaling must be carried out for the CDR values anyway, since not all simulated vehicles generate connection records. Looking at the scaling factors however, shows that there is no linear relation between the two arrays. While the gross of traffic should travel on the north - south axis, the highest count in the CDR data occurs in west - east direction. The range of the theoretical scaling factors is too big to assume a meaningful scaling possibility.

Table 4.3: CDR based OD estimation for the *Original Everything* data set

	Sensor [Veh/h]	CDR [Veh]	Theor. Scaling Factor
North - South	408	28	43.755
South - North	476	86	16.606
East - West	178	38	14.053
West - East	185	107	5.184

Further trials with other data sets do not lead to more convincing results. Two of them are presented in this paragraph, first an OD estimation of the *highway* data sets and second of the *residential* ones. The highway sets are very linear on the north - south axis. Their OD pairs are thus expected to be focused on these directions. This is not the case though. Since most vehicles that stay on the highway only connect to one cell during their commute, a direction cannot be identified for them with this technique. The residential data set on the contrary, includes slower vehicles that connect to several smaller cells. However, the precise OD generation fails simply because the movement directions are more randomized and the possibly involved centroids numerous.

4.2.5 Connection patterns

Regarding the long term use of CDR in traffic analysis, learning algorithms are of great importance. Apart from the standardized measurements that have been investigated in the previous algorithms, individualized patterns hold big potential. An algorithm that knows the shape of a data set during free flow conditions, will be able to recognize changes to this state better. Such algorithms need to be trained for a specific region. In the project, only one day is simulated and the learning effect hence is limited. Thanks to the different scenarios, it is still possible to gain and apply knowledge while analyzing. The part of the network mainly influenced by changing traffic conditions is the cell change procedure. The underlying algorithms for this process react to changed travel behavior. It is likely, that the same cell assignment is repeated for travelers under the same condition. This way, for one region, specific connection patterns can be linked to traffic conditions or even to road stretches. A pattern in this case means that vehicles connect to specific cells or to a sequence of cells. As soon as a certain cell choice becomes more popular in the region, the trained algorithm will know what traffic situation is the cause for this. It does so, by comparing the cell choices with its previously estimated patterns.

In a first analysis, it is examined in how far it is possible to determine the road choice of a vehicle based on its cell subscription. The Matlab function written for this purpose returns a matrix with one row for each existing pattern. The rows consist of the contained

Cell IDs and the sum of occurrences. This function is run with the *Original Everything* data set as input. In this case the *Original* Scenario is preferable, because it represents the most realistic traffic conditions. Thus, it is best to evaluate the potential of connection patterns for traffic research. In total, there are 113 different connection variations. They all consist of either one or two Cell subscriptions. The most commonly used patterns can be found in table 4.4. The first column of the table shows the ID of the cell related to each. The most popular connection patterns only consist of one cell subscription inside the simulation area. Thus, the pattern can be clearly identified by a single cell ID. The table is ordered by the number of occurrence within the *Everything* data set. It lists the most popular connections sequences in a descending order.

Table 4.4: Occurances of connection patterns in the *Original Scenario*

Pattern Cell ID	Everything Count	Highway Count	... and Ramps Count	Residential Count
13232	2688	77	1084	484
50292	1511	44	178	1424
44923	387	0	94	86
13190	366	374	387	4
47822	356	40	142	197
13161	280	45	286	0
47920	248	1	72	85
23050	242	320	0	0
23052	231	70	155	25
43093	212	143	252	0
13200	207	28	127	0
13251	188	25	19	225
44961	165	97	195	0
13420	160	87	134	40
13192	155	158	153	0
45131	137	26	168	0
23001	118	10	128	0
13402	114	121	92	0
45660	105	0	27	0

Next, the function is run again for all other data sets of the *Original* scenario. In the resulting matrices, some of the patterns from the first run are present again. Table 4.4 cross references the related counts with the previously identified common patterns. This way the counts for the same pattern can be compared in different data sets. It can hence be identified how big a part the single groupings play in each pattern. An outstanding example is the pattern in row 2 including cell 50292. It is the second most common connection for the whole area. The other counts show that this popularity almost exclusively results from residential travelers. Vice versa, any subscriber connected to 50292 can be assumed to drive on a residential road. The opposite can be seen from cell 23050 in row 8. The cell has the only high counts in the *Everything* and the *Highway* data set, while the rest of the counts are 0. Thus, it can be assumed that in both data sets, it has all its

subscribers from the highway.

In general, it can be concluded that an identification of the used link is possible in this simplified simulation environment. Once this is done, the investigation may be continued in greater detail for the single link. The application for connection patterns in the CDR data of single roads is to identify changes in the traffic situation. When the popularity of subscription patterns changes, a change in average speed of the travelers can be assumed. A relation between the vehicles' average speed and the cell choice has already been implied by the results presented in chapter 4.2.2. Thus, changed connection sequences can be used to spot congestion when compared with known patterns. Therefore, an initial state for comparison is needed. The best scenario to provide such is the *Free flow* scenario. Due to its uniform traffic movement, the highway data set is chosen as a simple example. In this data, the cell connection sequence for each vehicle is extracted. The results show that the vast majority of the connections is handled by the two cells 13190 and 23050. The first one takes most northbound travelers and the second the southbound ones. Their range is big and hence no further handovers are performed during the simulation. Those are the most important patterns for identifying free flow highway traffic. Figure 4.6 presents a plot of the sum of connections to those cells over time from the *Free flow Highway* data set.

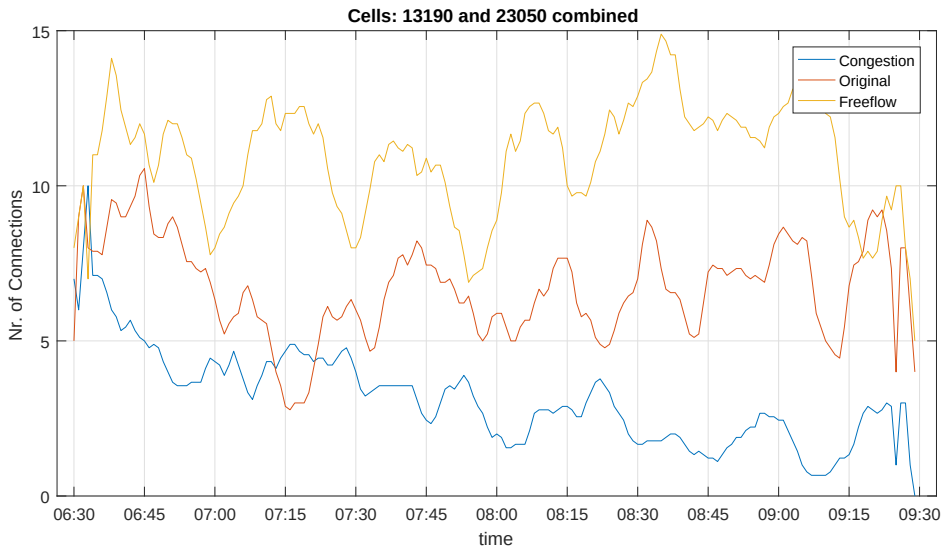

Figure 4.6: Popularity of free flow related connection patterns in the *Highway* data sets

Figure 4.6 also includes the corresponding plots for the other scenarios. From them it can be seen how the cells are equally popular in the beginning, but decline as traffic becomes more dense. While the *Original* curve recovers in the end, the *Congestion* plot stays on a low level. This represents longer lasting and more severe congestion in the latter scenario. Again the assumed potential of connection patterns can be approved even in such a limited environment where most of them only consist of one cell.

Chapter 5

Conclusion

A discussion of the results and the project specification gives a summary of what has been done. Furthermore, the chapter justifies the procedure and names its limitations. Answers to the research questions, presented in chapter 1.2 are given and the extend of their clarification discussed. All knowledge that is gained on the topic and all new research that can be based on it, is summarized. It is used to motivate a future outlook on upcoming and recommended research in the area of CDR data in traffic planning.

5.1 Discussion

This thesis describes the generation of synthetic mobile phone connection records from a microscopic simulation model. It follows a step by step approach throughout the whole process that is necessary to fulfill this task. Initially, an appropriate simulation area that contains a wide range of infrastructure and has available input data, is selected. Next, a simulation tool is chosen and the model to represent the geographical research area in it, built. Due to the software's specifications, simplifications of the reality are necessary. The focus therewith is kept on the project's requirements and the modeling parts that are crucial to them. A similar proceeding is applied regarding the input data for the simulation. Sensor data is only available for parts of the network and contains several errors and wrong counts. Regarding what is important for the experiments output, the input data is adjusted in details and partly assumed to make the simulation perform well. The cellular network overlay that is imported as simplified dots into the simulation relies on open source data of cell location estimates. However, the data is generated by a big community and contains a lot of samples. It is more focused on how a mobile network is perceived by the users than how it is actually constructed. This perspective has advantages regarding its application. The estimated cell ranges for example, are verified by users who were connected to cells in this distance.

While the model is running, an addition to the standard procedure is executed. By using the Aimsun API, the extension can interact with the current simulation in both directions and hence collect data about the simulation vehicles' trajectories. Within the API extension, the positions of a pre-selected group of vehicles are extracted on a regular basis. These positions are used as input for a mobile connectivity model that assigns

each of them to an appropriate cell. The range of existing algorithms to assign cells to subscribers is huge and they vary greatly in complexity and in the parameters they are based on. The algorithm used in the project does not attempt to be applicable in reality, but rather aims to simulate a valid cell distribution within the scenario. To some degree, the data is actively scrambled to mirror the structure of records generated through random calls. All valid steps of the experimental procedure are carried out using Aimsun. To reproduce the data with specifications, no other tool is necessary. This approach raises the utility regarding a reuse of the developed procedures. The representation of the cellular network in the program is rather simple. Models of propagation and advanced cell shapes are some parts of cellular networks that can not be represented in the project, as it was designed.

Throughout the data generation, simplifications and assumptions have to be made and there are a lot of different models involved in the process. However, the procedure is at every point focused on the study purpose and ensures that the models deliver meaningful results. All parts of the study are based on the knowledge gained through scientific literature research and work experience. Debugging and fault analysis have been carried out all along. Thanks to this effort, the data sets show a realistic behavior when being analyzed, as was done after their generation. Three different scenarios are implemented and thus three different series of data sets are produced. They differ in the demand input and either cause free flow conditions, congestion or an intermediate, as can be observed in reality. The different results for those scenarios are helpful to evaluate the results' sensitivity regarding traffic changes. Besides, additional data sets in each of the scenarios include only a selection of links to generate data from. This way, the influence of different road types on the results can be evaluated.

The applied evaluation approaches on the one hand attempt to verify the results by reproducing the conditions as implied by the simulation's demand input. On the other hand the algorithms themselves are being investigated to find out which measure is best to deduct information from the call records. The first conducted analysis regarding the total system load at each time of the simulation, unveils some differences between the scenarios. It is focused on the density of traffic and thus supposed to react on congestion. However, this approach only works for parts of the data sets and hence is not reliable. Furthermore, the total system load under real world conditions can be influenced by many factors. A fluctuation in its value does not necessarily have to mean an increase in vehicle traffic. The next two algorithms are more related to typical measurements for cellular networks. The data sets and scenarios are compared regarding the average radius of the subscribed cells. Network architecture causes faster travelers to connect to bigger cells, thus conclusions about their speeds can be drawn from the average cell size. A comparison of the different data sets within one scenario shows that this assumption holds. Cars on the highway are evidently faster than those on residential streets. Their average cell size is notably bigger, too. Traveling users can, following this approach, be separated from the noise data existing in real data sets. Specifically, the fact that cells of a radius over 9000 meters are primary used by highway travelers is of particular interest. In this way it is possible for the highway data to be filtered from the rest. Furthermore, by monitoring when those cells become more idle, congestion on the highway can be

spotted. This analysis however is limited to urban areas with multiple cellular layers. Additionally, the data sets are evaluated regarding the cell dwell time. This measurement targets the vehicles' speed. The slower a traveler becomes, the more time he spends inside the same cell. The analysis proofs that increased traffic in the *Congestion* data set can be effectively spotted. The dwell time is fast in its reaction to changes, since vehicles that are connected to a big cell and then have to slow down due to arising congestion spend especially much time in their cell. Later, when they are handed over to smaller, better fitting cells, the dwell time normalizes again.

A rather common implementation of CDR data in traffic engineering is to use it for generating OD matrices. The OD pairs are deducted from the vehicles' movement directions. Those directions can be obtained by following the travel routes based on the centers of the cells they are assigned to. For this study however, the attempt to reproduce the OD matrix from the data was not successful. The fact that many vehicles are just connected to one cell during the simulation, makes it hard to follow a route. Even for the rest, the localization is not precise enough for the small simulation area to generate meaningful travel directions from it. One approach that could potentially lead to a successful OD matrix estimation is through connection patterns. Two vehicles on the same route with the same traffic situation will most likely be connected to the same cells. Thus, common movement corridors can be identified from the specific sequence of cells they are connected to. The potential of this approach is proven by the successful separation of the main traffic streams in the *Original Scenario*. Connection pattern based algorithms are even able to learn over time. This has been simulated by identifying connection sequences on the highway under free flow conditions first. The most common sequences are then used as patterns. When monitoring these patterns under more congested conditions, a clear decline in their popularity can be observed. The longer an analysis like this is run, the more individual patterns can be recognized. Hence, more traffic situations can be identified.

Many of the algorithms tested here in a simple form, have the potential to create valuable traffic monitoring data. Since they all rely on different relations between cellular signaling and traffic situation, a combined application of them adds additional precision. To evaluate the algorithms closely, the procedures presented in this thesis are a good way to create more samples of connection records. The big advantage of those is, that the traffic condition they were created under are perfectly known and can be adjusted as desired. A compromise of CDR generation from microscopic simulation models is the size of the geographic area. Generally, bigger areas have advantages regarding the possible applications of the generated data. However, microscopic models are limited in size. Complexity and computational effort starts to grow significantly with increasing size of the model. This trade of will always limit the usability of synthetic CDR data.

5.2 Recapitulation of the research questions

Before starting the work on the project, its aims have been specified. One of them was to find a way to generate synthetic, standardized CDR from microscopic simulation models. To increase the reusability of the approach, as many parts of the procedure were to be standardized and applicable for multiple scenarios. This requirement has been achieved by several means. The experiment can be repeated for any microscopic simulation model created with the software Aimsun. The design of the model does not influence the extensions made. Different vehicle classes and public transport plans can be handled by the model as well. From the database of Open cell ID, it is possible to import cell center locations from all around the world. The community offers the widest range of different cell locations available. Thus, the models adaptation to input data from that source makes it widely usable. The procedure how to correctly filter the relevant cells from the worldwide data set is given in the report. Cell data from other sources may be used for the data generation as well. However, it will require some transformation to make it fit the requirements. The script used to extract the CDR from the simulation can be run in Aimsun directly and thus is not influenced by compatibility to other software. It is not possible to use it with another software or with a macroscopic/mesoscopic simulation study. The first intention of the project, to create a general approach for the generation of CDR from a microscopic simulation has been achieved.

Along with it, research questions concerning the analysis of the created output of the experiment have been formulated. It was intended to investigate how CDR data, collected in an urban region, can give information about the current traffic state on its road network. The first sub question emerging from this, was in how far a fast traveler could be distinguished from a slow one. This topic is addressed by chapter 4.2.3. It compares the two indicators of cell dwell time and number of records per vehicle. It is concluded that the cell dwell time gives a faster and stronger indication of a change in traveling speed. In addition, the size of the cells travelers are subscribed to gives an indication of their speed. Chapter 4.2.2 describes this relation on a system wide scale. Finally, connection patterns can be applied as described in chapter 4.2.5 to recognize traffic states that have been observed in the past. The size of the subscribed cell and the recognition of connection patterns can also be used to filter traveling users from a diverse CDR data set. This either works by selecting only the users connected to large cells or by following their movement through the sequence of their connections. By combining the previously mentioned techniques, traveling users can first be identified and later examined regarding changes in their behavior.

Chapter 4.2.1 examines the question, how to observe a changing density from CDR data. The approach to count the generated CDR entries within the research area makes changing density visible in the synthetic data set. Real data will though first have to be filtered for traveling users by the previously described means. In how far it is possible to directly derive demand changes from unfiltered CDR cannot be found out with the given data. Finally the project investigated the opportunities given by CDR data to distinguish OD pairs and route choices. This experiment has been conducted for bigger areas before, but not for an area as small as the modeled region of this project. Two approaches have

been compared in the chapters 4.2.4 and 4.2.5. The first of them attempts to distinguish some of the model's travel directions by the positions of the subscribed cells along a vehicle's path. The second assigns connection patterns to major links and recognizes them in other data sets. The results obtained in chapter 4.2.5 are much more promising. Based on such learning algorithms it is possible to read OD matrices from CDR data, even in a small geographic area.

5.3 Future outlook

The project in this thesis is a step towards understanding the structure of CDR data and its relation to traffic situations better. It has profited from the knowledge about numerous applications of such data in earlier research projects and some commercial purposes. At the end of the project, more projects can utilize the outcomes of this work and continue the research. There are several ways that the work can be carried on from here and a selection of them is to be presented. There is more room for experiments using the tools developed for the project. The time and resources of a university thesis are limited and so the efforts had to be focused. The possibility to test different research areas and more microscopic simulation models are numerous. Thanks to Open Cell ID, there is cellular network layouts available for most regions of the world. Of special interest will be to try a large stretch of highway in a rural area. The different shape of the network will require new means of analysis. Furthermore, a more simply structured network can be modeled in a wider range. This way, more advanced connection patterns should be observable. Thanks to a better quality of the cell location data in the United States of America, experiments in one of their urban regions are of interest too. The American network providers publish the locations of their antennas and their specifications. This can increase the precision of the research and lead to new conclusions. In order to make the models work under those new conditions, adjustments to them are recommended. More detailed cell tower location data, requires a more advanced mobile connection model. It may for example pay more attention to the shape of sectored cells or include the possibility of vertical cell changes during the simulation. Apart from this, different parameters for the call likelihood model can be tested and the effects evaluated. Later versions of Aimsun include the possibility to model pedestrian traffic. Including pedestrians in the models opens up a new challenge to filter the data and to identify specific travel modes and route choices. Additionally, a large number of pedestrians would generate a big amount of noise data that brings the synthetic data sets closer to real conditions. However, pedestrian models are hard to handle and typically include only small areas, too small to cause cell changes. Future research will have to find a way to overcome this problem. A smaller effort to extend the model is set by adding public transport lines to the system, as is also supported by Aimsun. A large number of people traveling together in one vehicle, will generate very similar connection patterns that might be identifiable from the whole.

Next to these extensions of the projects model, ongoing research also means to use the results generated by the project. A selection of data analysis methods has been pre-

sented. However, that selection is not exhausting the possibilities for more advanced and diversified approaches. Further research can utilize the tools and results of the project to motivate extended analysis of the results. Especially the topic of learning algorithms based on pattern recognition is believed to hold greater potential. On a long run perspective, testing the assumptions and theses on a real world data set is desirable. Synthetic data can never claim the same level of detail and advanced entropy as real one. The challenges for the algorithms will hence increase and will lead to their further improvement. Along all ongoing research the aim of the project has to be kept in mind. It is focused on producing CDR data samples and its models are exclusively adjusted to this purpose. Any future researcher has to bear this in mind when utilizing its tools for his/her own projects.

Bibliography

[1] Airsage. Nationwide Commute Report. Technical report, 2015.

[2] R. Akçelik. A review of gap-acceptance capacity models. *29th Conference of Australian Institutes of Transport ...*, (July):5–7, 2007.

[3] M. Alatise, M. Mzyece, and A. Kurien. A Handover Scheme for Mobile WiMAX Using Signal Strength and Distance. Technical report, Department of Electrical Engineering/French South African Institute of Technology (F'SATI) Tshwane University of Technology, 2009.

[4] L. Alexander, S. Jiang, M. Murga, and M. C. González. Origin-destination trips by purpose and time of day inferred from mobile phone data. *Transportation Research Part C: Emerging Technologies*, 58:240–250, 2015.

[5] American Association of State Highway and Transportation Officials. AASHTO Guidelines for Traffic Data Programs. 2009.

[6] C. Anoniou, R. Balakrishna, and J. Barcelo. *Fundamentals of traffic simulation*, volume 145. 2010.

[7] S. Antipolis. Digital cellular telecommunications system (Phase 2+); Radio subsystem link control (GSM 05.08 Version 5.1.0), 1996.

[8] K. Ayyappan and P. Dananjayan. Rss Measurement for Vertical Handoff in Hetrogeneous Network. *Theoretical and Applied Information Technology*, 4:989–994, 2008.

[9] J Barcelo, E. Codina, J. Casas, J. L. Ferrer, and D. Garcia. Microscopic Traffic Simulation : A Tool for the Design , Analysis and Evaluation of Intelligent Transport Systems. *Journal of Intelligent and Robotic Systems*, 41(2-3):173–203, jan 2004.

[10] M. Ben-Akiva, M. Ramming, and S. Bekhor. Route Choice Models. In *Human Behaviour and Traffic Networks*, pages 23–45. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[11] M. Berlingerio, F. Calabrese, G. Di Lorenzo, R. Nair, F. Pinelli, and M. Sbodio. AllAboard: A system for exploring urban mobility and optimizing public transport using cellphone data. *Lecture Notes in Computer Science*, 8190 LNAI(PART 3):663–666, 2013.

[12] M. Bierlaire. Route Choice Models : Introduction and Recent Developments, 2003.

[13] A. Blank. Traffic Counts and Traffic Surveys: Https://people.hofstra.edu/geotrans/eng/methods/ch9m2en.html.

[14] M. Brackstone and M. McDonald. Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour*, 2(4):181–196, 1999.

[15] K. Brown. Q&A with Kenneth Brown, Founder & CEO, Vehicle Occupancy Detection Corporation - Toll Roads News: http://tollroadsnews.com/news/qa-with-kenneth-brown-founder–ceo-vehicle-occupancy-detection-corporation, 2014.

[16] F. Calabrese and G. Lorenzo. Estimating Origin- Destination Flows using Mobile phone Location Data. *Cell*, 10:36–44, 2011.

[17] CBS News. Super Bowl 50: San Jose plans to track cellphone data - CBS News: Http://www.cbsnews.com/news/san-jose-will-track-cellphones-during-super-bowl-50/, 2016.

[18] J. Chambers. Cisco System's CEO John Chambers Features VODC's Technology In His Opening Address At Oracle Open World âĂŞ Vehicle Occupancy Detection: http://vehicleoccupancydetection.com/cisco-systems/, 2014.

[19] B. Ciuffo, V. Punzo, and M. Montanino. Thirty Years of Gipps' Car-Following Model. *Transportation Research Record: Journal of the Transportation Research Board*, 2315:89–99, 2012.

[20] S. Çolak, L. Alexander, B.G. Alvim, S. R. Mehndiretta, and M. C. Gonzalez. Analyzing Cell Phone Location Data for Urban Travel: Current Methods, Limitations and Opportunities. *TRB 2015 Annual meeting*, pages 1–17, 2015.

[21] Ericsson. Radio Resource Management 1., 2015.

[22] Federal Highway Administration. *Traffic Monitoring Guide*. U.S. Department of Transportation, Washington, 2013.

[23] H. T. Fritzsche. A model for traffic simulation. *Traffic Engineering & Control*, 35(5):317–321, 1994.

[24] K Fujimoto. *Mobile antenna systems handbook*. Artech House, Boston, third edition, 2008.

[25] V. Garg. *Wireless Communications and Networking*. 1st edition, 2007.

[26] P. G. Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B*, 15(2):105–111, 1981.

[27] P.a. Gonzalez, J.S. Weinstein, S.J. Barbeau, M.a. Labrador, P.L. Winters, N.L. Georggi, and R. Perez. Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks. *IET Intelligent Transport Systems*, 4(1), 2010.

[28] Google. Solna - Google Maps: https://www.google.se/maps/place/Solna/, 2016.

[29] S. Hoogendoorn and P. Bovy. State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 4(215):283–303, 2001.

[30] L. F Huntsinger and D. K. Ward. Using Mobile Phone Location Data to Develop External Trip Models. In *Transportation Research Board Annual Meeting*, 2015.

[31] M. S. Iqbal, C. F. Choudhury, P. Wang, and M. C. González. Development of origin-destination matrices using mobile phone call data. *Transportation Research Part C: Emerging Technologies*, 40:63–74, 2014.

[32] H. Y. Jeong, M. S. Obaidat, N. Y. Yen, and J. J. Park. *Advances in Computer Science and its Applications*, volume 279 of *Lecture Notes in Electrical Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[33] F. Johansson. Car-following models, 2015.

[34] R. T. Juang, H. P. Lin, and D. B. Lin. An improved location-based handover algorithm for GSM systems. *IEEE Wireless Communications and Networking Conference, 2005*, 3:1371–1376, 2005.

[35] J. K. Kim, J. Kim, and M. Chang. Lane-changing gap acceptance model for freeway merging in simulation. *Canadian Journal of Civil Engineering*, 35(3):301–311, mar 2008.

[36] R. Kitamura and M. Kuwahara. *Simulation Approaches in Transportation Analysis: Recent Advances and Challenges*. Springer Science & Business Media, 2006.

[37] L-com. 3Sektor-omni antenna: http://www.l-com.com/multimedia/diagrams/d_HK913-120_1.gif, 2016.

[38] G. Leduc. Road Traffic Data : Collection Methods and Applications. *EUR Number: Technical Note: JRC 47967*, JRC 47967:55, 2008.

[39] J. B. Lesort, A. Nuzzolo, F. Russo, and A. Vitetta. A Modified Logit Route Choice Model Overcoming Path Overlapping Problems. Specification And Some Calibration Results For Interurban Networks. In *Transportation and Traffic Theory*, pages 697–711. Pergamon, 1996.

[40] E. Letouzé and P. Vinck. The Law, Politics and Ethics of Cell Phone Data Analytics. *Data-Pop White Paper Series*, (April), 2015.

[41] Libelium. Detecting iPhone and Android Smartphones by WiFi and Bluetooth [Cellular - Mobile - Hand Phone Detection] | Libelium: http://www.libelium.com/products/meshlium/smartphone-detection/.

[42] M. J. Lighthill and G. B. Whitham. On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 229(1178):317–345, 1955.

[43] P. Marichamy, S. Chakrabarti, and S.L. Maskara. Performance evaluation of handoff detection schemes. *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, pages 643–646, 2003.

[44] A. Markopoulos, P. Pissaris, and S. Kyriazakos. Efficient location-based hard handoff algorithms for cellular systems. *Networking*, 3042:476–489, 2004.

[45] S. A. Mawjoud. Simulation of Handoff Techniques in Mobile Cellular Networks. *Al-Rafidain Engineering*, pages 69–80, 2005.

[46] G. Mitchell. *The Practice of Operational Research*. Wiley, 1993.

[47] Y. Montjoye and Z. Smoreda. D4D-Senegal : The Second Mobile Phone Data for Development Challenge. (1):1–11, 2014.

[48] Prof S. Nandakumar. Traffic Driven & Received Signal Strength Adaptive Handoff Scheme. *International Journal*, 21(6):30–35, 2011.

[49] J. A. Nordstrand. Localizing Cell Towers from Crowdsourced Measurements Supervisor :. 2015.

[50] J.J. Olstam and A. Tapani. Comparison of Car-Following Models. *Transportation Research Record*, 1678(1):116–127, 2004.

[51] Open Cell ID. Menu map view - OpenCellID wiki: http://wiki.opencellid.org/wiki/Menu_map_view#database, 2016.

[52] J. Ortuzar and L. G. Willumsen. *Modelling Transport*. John Wiley & Sons, Ltd, 2011.

[53] S. Pahal and B. Singh. Performance Evaluation of Signal Strength and Residual Time based Vertical Handover in Heterogeneous Wireless Networks. 31(1):25–31, 2014.

[54] M. Rahmani and H. Koutsopoulos. Requirements and potential of GPS-based floating car data for traffic management. *IEEE Conference on Intelligent Transportation Systems, Proceedings*, pages 730–735, 2010.

[55] M. Rani, S. Behara, and K. Suresh. Comparison of Standard Propagation Model (SPM) and Stanford University Interim (SUI) Radio Propagation Models for Long Term Evolution (LTE). Technical report, Department of ECE, Chaitanya Engineering College, Visakhapatnam, A.P. INDIA, 2012.

[56] D. C. Sati. Application of Soft Computing Techniques for Handoff Management in Wireless Cellular Networks. (6):1–6, 2012.

[57] D. Sinclair. Cellphone Data Could Help in Developing Transportation Plan | News | thepilot.com, 2013.

[58] Ten and Two. Origin-Destination - Data Collection Software: http://www.l2datacollection.com/pages/origin_destination.html, 2016.

[59] J. L. Toole, S. Colak, B. Sturt, L. P. Alexander, A. Evsukoff, and M. C. Gonzalez. The path most traveled: Travel demand estimation using big data resources. *Transportation Research Part C: Emerging Technologies*, 58:162–177, 2015.

[60] M. Treiber and A. Kesting. *Traffic Flow Dynamics. Data, Models and Simulation.* Springer Berlin Heidelberg, 2013.

[61] D. Triantafyllos. Exporting probe data using Aimsun micro API - TSS-Transport Simulation Systems | TSS-Transport Simulation Systems: https://www.aimsun.com/exporting-probe-data-using-aimsun-micro-api/, 2015.

[62] TSS. Aimsun 7 Scripting Manual. *Methods*, (November), 2011.

[63] TSS. Aimsun 7 Dynamic Simulators User ' s Manual October 2012. *Transport Simulation Systems*, 7(October):480, 2012.

[64] D. Veeneman. Cellular Signalling. *Monitoring times*, (December), 1996.

[65] M. Wall. Ebola: Can big data analytics help contain its spread? - BBC News, 2014.

[66] Y. Wang, G. Homem, and E. Romph. National and Regional Road Network Optimization for Senegal Using Mobile Phone Data. Technical report, Department of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2014.

[67] J. White, J. Quick, and P. Philippou. The use of mobile phone location data for traffic information. In *Road Transport Information and Control, 2004. RTIC 2004. 12th IEE International Conference on*, pages 321–325, 2004.

[68] E. Williams. Latitude/Longitude Distance Calculator: http://www.nhc.noaa.gov/gccalc.shtml, 2016.

[69] M. D. Yacoub. Gsm. In *Wireless Technology, Protocols, Standards, and Techniques*, number 121. 2001.

[70] Michel Daoud Yacoub. Cellular Principles. In *Wireless Technology, Protocols, Standards, and Techniques*, chapter 2. 2001.

[71] G. Yan. Finding a Goldmine in Cellular Data: https://www.accenture.com/us-en/blogs/blogs-finding-a-goldmine-in-cellular-data-a-new-opportunity-for-telecom-in-the-big-data-era, 2014.

[72] Y. Zhang, X. Qin, S. Dong, and B. Ran. Daily O-D Matrix Estimation Using Cellular Probe Data. *Transportation Research Board 89th Annual Meeting*, 2010.

[73] M. M. Zonoozi. Handover in Digital Cellular Mobile Communication Systems. (March), 1997.

Appendix

Table A.1: OD matrix [Veh/h], used from 6:45-7:00am

O/D	E18 west	Solna center	Frösundaleden	E4 south	E4 north	E18 east	Totals
E18 west	0	25	6	140	123	185	480
Solna Center	26	0	22	54	52	26	179
Frösundaleden	9	23	0	19	18	9	79
E4 south	135	30	9	0	395	135	703
E4 north	149	27	8	411	0	149	745
E18 east	185	107	5	149	143	0	589
Totals	504	213	51	773	731	504	2775

Table A.2: OD matrix [Veh/h], used from 7:00-7:15am

O/D	E18 west	Solna center	Frösundaleden	E4 south	E4 north	E18 east	Totals
E18 west	0	24	6	133	126	180	469
Solna Center	25	0	21	51	52	25	175
Frösundaleden	9	22	0	18	19	9	77
E4 south	137	31	9	0	401	137	715
E4 north	141	26	8	390	0	141	706
E18 east	180	104	5	141	145	0	575
Totals	492	208	50	733	743	492	2718

Table A.3: OD matrix [Veh/h], used from 7:15-7:30am

O/D	E18 west	Solna center	Frösundaleden	E4 south	E4 north	E18 east	Totals
E18 west	0	26	7	141	137	193	503
Solna Center	27	0	23	54	57	27	188
Frösundaleden	10	24	0	19	20	10	83
E4 south	149	34	10	0	437	149	779
E4 north	150	28	8	414	0	150	750
E18 east	193	112	5	150	158	0	618
Totals	529	223	53	779	809	529	2921

Table A.4: OD matrix [Veh/h], used from 7:30-7:45am

O/D	E18 west	Solna center	Frösundaleden	E4 south	E4 north	E18 east	Totals
E18 west	0	26	7	141	140	195	509
Solna Center	27	0	23	54	59	27	190
Frösundaleden	10	24	0	19	21	10	84
E4 south	153	34	10	0	448	153	798
E4 north	150	28	8	414	0	150	750
E18 east	195	113	5	150	162	0	625
Totals	534	225	54	778	829	534	2955

Table A.5: OD matrix [Veh/h], used from 7:45-8:00am

O/D	E18 west	Solna center	Frösundaleden	E4 south	E4 north	E18 east	Totals
E18 west	0	26	7	138	145	195	511
Solna Center	27	0	23	53	61	27	191
Frösundaleden	10	24	0	19	21	10	84
E4 south	158	36	11	0	464	158	828
E4 north	146	27	8	404	0	146	731
E18 east	195	113	5	146	168	0	627
Totals	536	226	54	759	860	536	2971

Table A.6: OD matrix [Veh/h], used from 8:00-8:15am

O/D	E18 west	Solna center	Frösundaleden	E4 south	E4 north	E18 east	Totals
E18 west	0	27	7	139	149	198	520
Solna center	28	0	23	53	62	28	195
Frösundaleden	10	24	0	19	22	10	85
E4 south	162	37	11	0	476	162	848
E4 north	148	27	8	408	0	148	739
E18 east	198	115	6	148	172	0	638
Totals	546	230	55	768	881	546	3025

Table A.7: OD matrix [Veh/h], used from 8:15-8:30am

O/D	E18 west	Solna center	Frösundaleden	E4 south	E4 north	E18 east	Totals
E18 west	0	26	7	137	146	195	511
Solna Center	27	0	23	53	61	27	192
Frösundaleden	10	24	0	19	22	10	84
E4 south	159	36	11	0	467	159	832
E4 north	146	27	8	403	0	146	729
E18 east	195	113	5	146	169	0	628
Totals	537	226	54	757	865	537	2976

Table A.8: OD matrix [Veh/h], used from 8:30-8:45am

O/D	E18 west	Solna center	Frösundaleden	E4 south	E4 north	E18 east	Totals
E18 west	0	26	6	134	142	190	497
Solna Center	27	0	22	51	59	27	186
Frösundaleden	10	23	0	18	21	10	82
E4 south	155	35	10	0	453	155	808
E4 north	142	26	8	392	0	142	710
E18 east	190	110	5	142	164	0	610
Totals	522	220	53	737	839	522	2893

Table A.9: OD matrix [Veh/h], used from 8:45-9:00am

O/D	E18 west	Solna center	Frösundaleden	E4 south	E4 north	E18 east	Totals
E18 west	0	23	6	118	132	172	451
Solna Center	24	0	20	45	55	24	169
Frösundaleden	9	21	0	16	20	9	74
E4 south	144	33	10	0	423	144	753
E4 north	125	23	7	347	0	125	628
E18 east	172	99	5	125	153	0	554
Totals	474	199	48	652	782	474	2630

Table A.10: OD matrix [Veh/h], used from 9:00-9:15am

O/D	E18 west	Solna center	Frösundaleden	E4 south	E4 north	E18 east	Totals
E18 west	0	22	5	111	125	161	424
Solna Center	23	0	19	43	52	23	159
Frösundaleden	8	20	0	15	18	8	70
E4 south	136	31	9	0	398	136	710
E4 north	118	22	7	326	0	118	590
E18 east	161	93	5	118	144	0	521
Totals	446	187	45	612	737	446	2473

Table A.11: OD matrix [Veh/h], used from 9:15-9:30am

O/D	E18 west	Solna center	Frösundaleden	E4 south	E4 north	E18 east	Totals
E18 west	0	22	5	111	127	163	429
Solna center	23	0	19	43	53	23	161
Frösundaleden	8	20	0	15	19	8	71
E4 south	139	31	9	0	407	139	726
E4 north	118	22	7	327	0	118	592
E18 east	163	94	5	118	147	0	527
Totals	451	190	45	614	754	451	2506

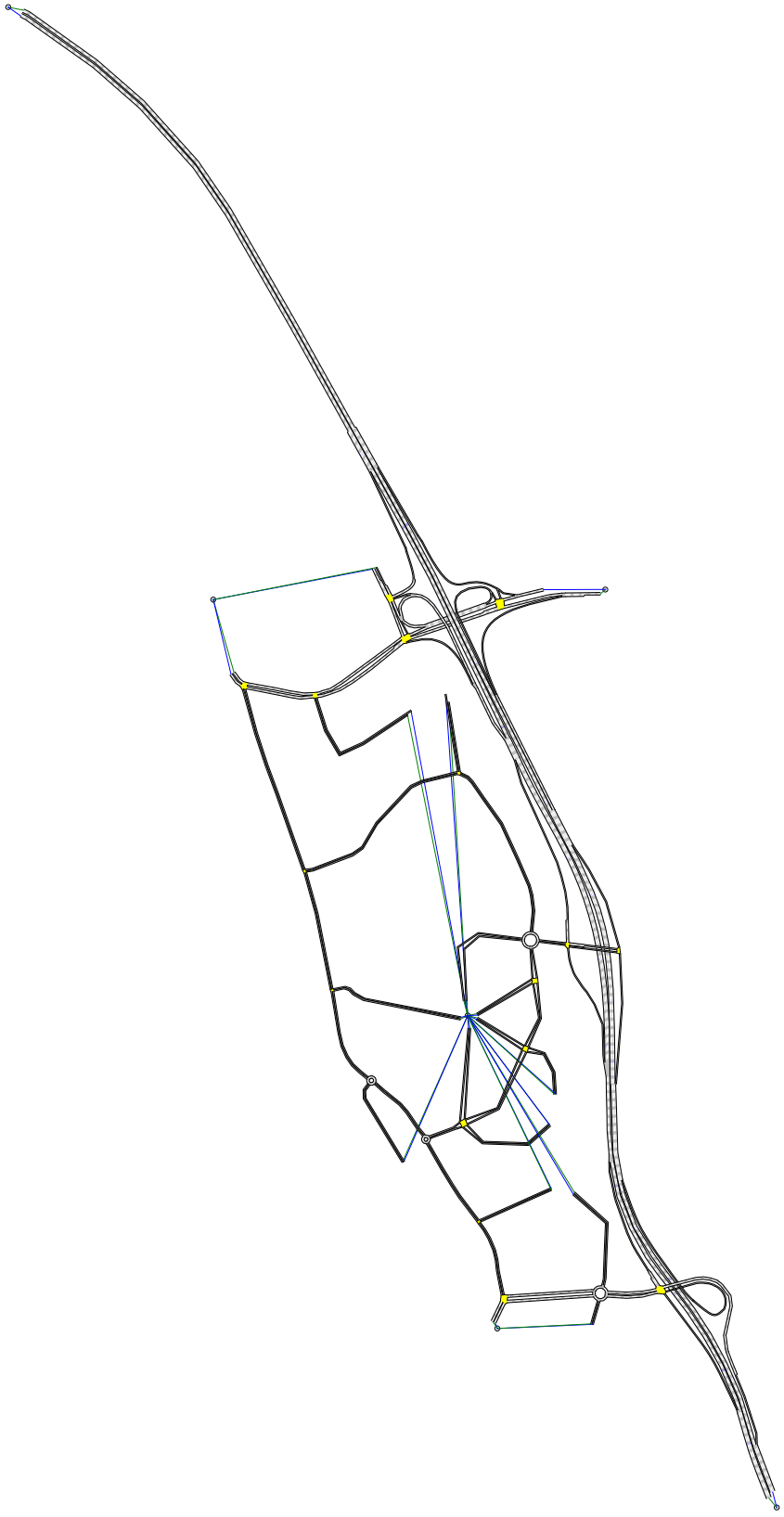

Figure A.1: The whole network as it is modeled in Aimsun

Python script for CDR generation

```
1  #For the Code to work the cells should be imported to Aimsun as Points. An
       External ID for each cell containing its radius is required. It is
       recommended to use Input Layer generated from OpenCellId-data.
2
3  from AAPI import *
4  import sys
5  from PyANGKernel import *
6  import datetime
7
8  Cycle=17              #Distinguish the timesteps for retreiving data from the
       simulation
9  relative=0.2            #the relative number of users receiving a call
10 SG=41692              #ID of the Grouping of sections
11 PG=41689              #ID of the Grouping of Cell towers
12 filename ='ProbeData_everything.csv'  #Filename of the output file
13
14 model = GKSystem.getSystem().getActiveModel()
15 Connectionrecord={}
16
17 def AAPILoad():
18   global filename
19
20   Results = open(filename, "w")
21   Results.write('Vehicle_ID,time,origin,Cell-ID \n')        #Headline for the
         Output file
22   Results.close()
23   return 0
24 def AAPIInit():
25   return 0
26 def AAPIManage(time, timeSta, timeTrans, acycle):
27   return 0
28
29 def AAPIPostManage(time, timeSta, timeTrans, acycle):       #time=time elapsed
       during current Experiment; time Sta=time+Start time; acycle=time difference
        between simulation steps
30   global Cycle, relative, filename, SG, PG, Connectionrecord        #Global
         variables that have been defined in the root
31   if time%Cycle<acycle*0.9:                    #time is almost evenly dividable
         by the cycle
32     timeformatted = str(datetime.timedelta(seconds=timeSta))    #display the
         time in a nice way
33     Results = open(filename, 'a')                #append new results to output
         file
34     sections = model.getCatalog().find(SG).getObjects()      #get all Objects
         from the sections grouping
35     points = model.getCatalog().find(PG).getObjects()      #get all objects from
          the pointsgroup as GKDPoint
36     Celltower = points[1]
37     for i in sections:
38       id = i.getId()
39       vehicles = AKIVehStateGetNbVehiclesSection(id,True)
40       for j in range(vehicles):                #loop to generate records for
           cars on the section
41         random = AKIGetRandomNumber ()
```

```
42          if random>relative:                    #represents the call likelihood
43            break
44          probe = AKIVehStateGetVehicleInfSection(id, j)
45          probeId = probe.idVeh
46          probePOS = GKPoint(probe.xCurrentPos, probe.yCurrentPos)
47          speedfactor = (probe.CurrentSpeed**2)+500
48          if probeId not in Connectionrecord:          #if the vehicle is
              regarded for the first time
49            dist = 999999.9                  #very high value
50            Connectionrecord[probeId] = 'Error'
51            for k in points:                   #loop checks all celltowers to find
                the closest
52              radius = k.getExternalId().toInt()
53              if 0.7<speedfactor/radius<1.3:          #cell choice based on multi
                  layer network model
54                pointPOS = k.getPosition()
55                if pointPOS.distance2D(probePOS)<dist:      #if this is the
                    closest tower so far
56                  dist= pointPOS.distance2D(probePOS)
57                  Celltower = k
58            Results.write("%i,%s,%s,%s \n"%(probeId, timeformatted,
                Connectionrecord[probeId], Celltower.getName()))
59            Connectionrecord[probeId] = Celltower
60          else:          #if there has been a connection to another Cell before
61            former = Connectionrecord[probeId].getPosition()
62            dist = former.distance2D(probePOS)
63            Celltower = Connectionrecord[probeId]
64            if dist<Celltower.getExternalId().toInt():      #If the former cell
                is still in range, prevent Handover
65              dist =0
66            for k in points:                   #loop checks all celltowers to find
                the closest
67              radius = k.getExternalId().toInt()              #gets the curent
                  cell's range
68              if 0.7<speedfactor/radius<1.3:          #cell choice based on multi
                  layer network model
69                pointPOS = k.getPosition()
70                pointPOS = k.getPosition()
71                if pointPOS.distance2D(probePOS)<dist:      #if this is the
                    closest tower so far
72                  dist= pointPOS.distance2D(probePOS)
73                  Celltower = k
74            Results.write("%i,%s,%s,%s \n"%(probeId, timeformatted,
                Connectionrecord[probeId].getName(), Celltower.getName()))
75            Connectionrecord[probeId] = Celltower
76
77      Results.close()
78    return 0
79
80  def AAPIFinish():
81    return 0
82  def AAPIUnLoad():
83    return 0
84  def AAPIEnterVehicle( idveh, idsection):
85    return 0
86  def AAPIExitVehicle( idveh, idsection):
87    return 0
```

```
88  def AAPIPreRouteChoiceCalculation(time, timeSta):
89      return 0
```

Matlab script for sensor data extraction

```
1  %This script is used  for averaging the flow and creating traffic states in 15
       min intervalls. The input flow is a 3D matrix created from the MMS Matlab
       tool by Rasmus Ringdal
2
3  clc;
4  a=0;
5  columnaverage=zeros(28,182);
6  rowaverage=zeros(28,14);
7  N=length(flow);      %up to the highest number of sensor (1206)
8
9  for i = 1:N
10     if sum(flow(:,i,:))>0   %check if there is data for this sensor
11        a=a+1;               %start a new row
12        b=1;                 %reset the counter for averaging
13        used =squeeze(flow(:,i,:)); %2D matrix for the current sensor
14        columnaverage(a,1)=i;   %write the sensor ID in the first column
15        rowaverage(a,1)=i;
16        M=length(used); %The no. of counts per day for one sensor (181)
17        for j = 1:M
18            u=used(:,j);      %monthly data for one sensor and one minute
19            v=sort(u(u~=0)); %sort all values !=0 from small to big
20            w=v(2:end-1);     %delete the smallest and the highest value
21            columnaverage(a,j+1)=round(sum(w)/length(w));     %put the average of
                  the remaining values as result
22            if j/b > 15
23                b=b+1;        %Every 15 steps, b is increased by 1
24            end
25            rowaverage(a,b+1)= rowaverage(a,b+1)+columnaverage(a,j+1);    %Sum up
                  15 averages each time
26        end
27     end
28 end
29
30 rowaverage=rowaverage(:,1:13);  %Cut the matrix to the correct size
31 rowaverage(:,2:end)=round(rowaverage(:,2:end)./15);  %Create the average flow
       per hour
32
33 %The scripts results in a 2D matrix (rowaverage) with one row per used sensor
       and one column per 15 min interval
```