A Machine Learning Approach to Understanding Transportation Mode Behavior

Koch, Thomas Hans Arjen

2022

document version
Publisher's PDF, also known as Version of record

Mobility and Accessibility in the Metropolitan Region of Amsterdam

A Machine Learning Approach to Understanding Transportation Mode Behavior

VRIJE UNIVERSITEIT

MOBILITY AND ACCESSIBILITY IN THE METROPOLITAN REGION OF AMSTERDAM

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. J.J.G. Geurts,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Bètawetenschappen
op maandag 16 mei 2022 om 11.45 uur
in een bijeenkomst van de universiteit,
De Boelelaan 1105

door

Thomas Hans Arjen Koch

geboren te Lelystad

promotor:	prof.dr. R.D. van der Mei
copromotor:	dr. E.R. Dugundji
promotiecommissie:	prof. dr. G.M. Koole
	prof. dr. E.T. Verhoef
	dr. E.J.E.M. Pauwels
	prof.dr. T. Bellemans
	dr. M. Snelder

CONTENTS

INTRODUCTION

This dissertation forms part of the North/South-line Impact Study, a four-year research project on the effects of the opening of a new metro line in Amsterdam connecting the north with the south of the city. Our main research question focuses on mobility and accessibility: *"What are the effects of the opening of the North/South metro line for the entire transportation system in the Greater Amsterdam Metropolitan region?"* In particular we examine how the new metro line affects: (1) changes in multi-modal travel behavior, (2) changes in the number of trips and travel distance per transportation mode, and (3) travel times and congestion for car and bicycle. We examine each of these three dimensions in turn, in the three parts of this thesis.

PART I: ACCESS TO THE AMSTERDAM METROPOLITAN REGION

In Part I, we examine the role of public transit in multi-modal travel behavior in the Greater Amsterdam Metropolitan region. We first consider changes in public transit access to the region while explicitly addressing changes in walk mode behavior to/from public transit. Subsequently we will consider changes in car access to the region and changes in parking behavior at garages near public transit hubs. In particular, chapters 1 and 2 consider how accessibility can be framed in terms of door-to-door travel time using public transport and study how this door-to-door travel time was affected by the new metro line. Additionally, we consider the impact of the frequency of public transit schedules. Chapter 3 forecasts off-street parking occupancy of garages in Amsterdam and studies changes in car parking before and after the opening of the North/South-line.

DOOR-TO-DOOR TRANSIT ACCESSIBILITY USING PARETO OPTIMAL RANGE QUERIES

To determine the changes in travel time door-to-door by public transport, *before* and *after* the Noord/Zuidlijn we looked at routing algorithms to calculate all travel options within a time range. This way, we could study the effect of frequencies on the average in-vehicle time combined with the average waiting time between services for time periods such as the morning rush hour. To calculate the travel time that was perceived if we include average wait time between each service, we used the rooftop method.

To determine how door-to-door travel time changed on public transport due to the Noord/Zuidlijn, we used routing algorithms to compute the travel time door-to-door. Specifically we look at all the travel options with in time range of two hours, to study the effect of frequency. In chapter 1 we look how travel times are distributed within the CBS neighborhood (buurt) as origin and destination. We show that in many cases the total travel time is not normally distributed but rather has a bi-modal or even tri-modal distribution. We also show how this originates from accessibility properties of public transit stops. A series of experiments show that the distribution for the travel time can

be accurately estimated using on the order of ten or a few dozens of locations sampled in each zone.

THE IMPACT OF PUBLIC TRANSIT FREQUENCIES ON DOOR-TO-DOOR TRAVEL TIMES

In chapter 2, we build on this study to estimate travel times between the 799 CBS neighborhood zones, by sampling 6693 addresses total. We sample for spatial diversity by adding the constraint that no sampled address can be located within 200 meters of an address that was already sampled within that zone. Subsequently, we add the weight of each address to the nearest addresses, with a weight function, allotting unit weight per residential address and weighting other buildings based on their surface area. Using the weighted average we build an interactive tool to determine the differences in travel time between periods before and after the Noord-Zuidlijn and various timetables along the COVID-19 pandemic.

SHORT-TERM FORECASTING OF OFF-STREET PARKING OCCUPANCY

In chapter 3, we look at the changes in occupancy in off-street parking locations and park and ride locations. We conduct an effect analysis into the influence of weather, events, parking tariffs and public transport attributes on the parking occupancy. The most influential factors are the event variables, presence of thunderstorms, average wind speed, temperature, precipitation, sunshine, and the addition of a public transport line. Parking tariffs do not significantly contribute to the model performance, which may be due to the lack of temporal variability in the parking tariffs of the examined parking locations.

PART II: TRIPS AND TRAVEL DISTANCE

In Part II we first forecast public transport ridership in the Greater Amsterdam Metropolitan region by applying neural networks using public transit smart card data collected by the fare system. We complete this part with a study on using neural nets to study transportation mode choice behavior across transportation modes using GPS electronic trace data collected by a mobile phone app within the Impact Study project.

FORECASTING PUBLIC TRANSIT RIDERSHIP USING NEURAL NETWORKS

In chapter 4 we use ridership data provided by public transit smart card (OV-chipkaart) data to build a framework for the efficient representation and analysis of both spatial and temporal dimensions of panel data, representing this data as a series of image-matrices. The relative performance of a set of machine learning techniques is examined, focusing on Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) neural networks. Furthermore Sequential CNN-LSTM, Parallel CNN-LSTM, Augmented Sequential CNN-LSTM are explored. All models are benchmarked against a Fixed Effects Ordinary Least Squares regression. Results showed that the forecasts produced by the Sequential CNN-LSTM model performed best and suggest that the proposed framework could be utilized in applications requiring accurate modeling of demand for public transport. The described augmentation process of Sequential CNN-LSTM could be used to introduce exogenous variables into the model, potentially making the model more

explainable and robust in real-life settings.

USING NEURAL NETWORKS TO PREDICT TRANSPORTATION MODE CHOICE BEHAVIOR

In chapter 5 we assess changes in traveler behavior that occurred after the opening of the Noord/Zuid-lijn we explored mode choice substitution patterns using panel data collected within the framework of the Impact Study. As it is known that artificial neural nets excel at complex classification problems, this paper aims to investigate an approach where the traveler's transportation mode is predicted through a neural net, trained on choice sets and user specific attributes inferred from the data. The method shows promising results. It is shown that such models perform better when it is asked to predict the choice of mode for trips which take place on the same underlying transportation network as the data with which the model is trained. This difference in performance is observed to be especially high for trips from and to certain areas that were impacted by the introduction of the North South line, indicating possible changes in behavioral patterns.

PART III: TRAVEL TIME AND CONGESTION

In Part III we look at various aspects of travel time and congestion. We start of with a study on finding the optimal points to place car/bicycle counters for traffic studies. Next we look at a comparison of approaches for time series forecasting of traffic flow rates around Amsterdam using car traffic counters. Finally we look at bicycle route choice. First we look how path complexity can help in understanding bicycle route choice and we conclude with a final study estimating bicycle route choice models.

THE OPTIMIZATION OF TRAFFIC COUNTER LOCATIONS IN MULTI-MODAL TRANSPORTATION NETWORKS

In chapter 6 we we investigated ways to optimize the placement and number of traffic counters used in multi-modal transportation analysis studies for motorized vehicles, bicycles and pedestrians. The goal is to strike a balance between using as few as possible traffic counters for economical efficiency and deploying more counters which could collect more data. By using shortest path algorithms to determine the paths between the centroids of statistical divisions, we derive from origin-destination matrices which traffic is flowing from where to where over which links in a multi-modal network. Using centrality measures such as betweenness, we determine the links in the transportation networks that capture the most useful traffic in terms of as much unique traffic as possible.

A COMPARISON OF APPROACHES FOR THE TIME SERIES FORECASTING OF MOTORWAY TRAFFIC FLOW RATE

In chapter 7 we looked at time series forecasting of the traffic flow rate for 41 different locations in and around Amsterdam, The Netherlands. Using TBATS, SARIMAX and LSTM models, among others, the traffic flow rate of these locations has successfully been modelled. These models may provide accurate predictions for the future flow rate, which may be useful for the identification of infrastructure bottlenecks and the scheduling of

maintenance. Considering the dramatic effects of the COVID-19 pandemic on the traffic flow rate, the inclusion of 2020 data with a number of external factors, could lead to an improvement of the results and the ability to model the future effects of the pandemic.

PATH COMPLEXITY AND BICYCLE ROUTE CHOICE SET ASSESSMENT

In chapter 8 we look at bicycle route choice and route complexity. Everyday route choices made by bicyclists are known to be more difficult to explain than vehicle routes, yet prediction of these choices is essential for guiding infrastructural investment in safe cycling. Building route choice sets to study bicycle route choice is a difficult task. Even including detailed attributes such as the number of left turns, the number of speed bumps, distance and other route choice properties we still see that choice set quality measures suggest poor replication of observed paths.

In this chapter we looked at how the concept of *route complexity* can help generate and analyze plausible choice sets in the demand modeling process. The complexity of a given path in a graph is the minimum number of shortest paths that is required to specify that path. *Complexity* is a path attribute which could potentially be considered to be important for route choice in a similar way. The complexity was determined for a large set of observed routes and for routes in the generated choice sets for the corresponding origin-destination pairs. The respective distributions are shown to be significantly different so that the choice sets do not reflect the traveler preferences, this is in line with classical choice set quality indicators. Secondly, we investigate often used choice set quality methods and formulate measures that are less sensitive to small differences between routes that can be argued to be insignificant or irrelevant. Such difference may be partially due to inaccuracy in map-matching observations to dense urban road networks.

LIMITATIONS OF RECURSIVE LOGIT FOR INVERSE REINFORCEMENT LEARNING OF BICYCLE ROUTE CHOICE BEHAVIOR IN AMSTERDAM

In chapter 8 we take a look at another method to understand bicycle route choice. Used for route choice modelling by the transportation research community, recursive logit is a form of inverse reinforcement learning, which avoids the necessity to create route choice sets. By solving a large-scale system of linear equations recursive logit allows estimation of an optimal (negative) reward function in a computationally efficient way that performs for large networks and a large number of observations. In this paper we review examples of recursive logit and inverse reinforcement learning models applied to real world GPS travel trajectories and explore some of the challenges in modeling bicycle route choice in the city of Amsterdam using recursive logit as compared to a simple baseline multinomial logit model with environmental variables.

We discuss conceptual, computational, numerical and statistical issues that we encountered and conclude with recommendation for further research. Finally we compare this work to a conventional choice model, estimating models using multi-nominal logit, mixed logit and mixed path size logit specifications. Our results show that cyclists have a highly stochastic behavior that are likely to prefer detours to drive over cycle-way infrastructure, near greener land-use and near water, and on less busy roads. Models such as mixed logit that can estimate the stochasticity perform best to capture this behavior.

I

PART I: ACCESS TO THE AMSTERDAM METROPOLITAN REGION

1

Door-to-door Transit Accessibility using Pareto-Optimal Range Queries

Public transit is a backbone for well-functioning cities, forming a complicated system of interconnecting lines each with their own frequency. Defining accessibility for public transit is just as complicated, as travel times can change every minute depending on location and departure time. With Pareto-optimal journeys it is possible to look beyond the earliest arrival times and also optimize for the shortest travel time, as travelers base their departure time on the start time given by their smartphone app, especially when service frequencies are low. By querying for all Pareto-optimal journeys in a time range, it becomes possible to get a grasp of what passengers see as their choice set when it comes to transit route choice. The Pareto-frontier contains only the fastest alternatives for reaching the destination, at all departure times, removing options where departing earlier from Vijzelgracht will lead to longer wait times at Noord to get on the same bus to Purmerend. Based on the averages of the Pareto-optimal journeys it should become possible to calculate more realistic skim matrices for traffic analysis zones, including reliability factors such as frequencies and the number of transfers. In this chapter, we calculate Pareto-optimal journeys in the area in and around Amsterdam, looking at how travel times are distributed and what factors impact them.

Koch, T., Knapen, L., & Dugundji, E. (2020). Door-to-door transit accessibility using Pareto-optimal range queries. Procedia Computer Science, 170, 107-114.

1

1.1. INTRODUCTION

Public transportation is an important travel mode that keeps cities liveable. Determining travel times for public transport alternatives is a more difficult task than for pedestrians, bicyclists and even cars. This has multiple reasons, starting with the fact that public transportation always involves another modality such as walking or cycling, meaning that public transportation accessibility is heavily dependent on the distance from the origin to the nearest location to board a transit vehicle and the distance between the destination and the nearest location to disembark transit. A main reason that analysis of public transportation is difficult is that we are dealing with a time-dependent network, as transit is an intricate system of buses, trains, trams, metros that drive in frequencies that change depending on the time-of-day and are affected by external factors such as traffic and weather. To compensate for these external factors, timetables often include extra time in the travel time and transfer times, so passengers will still be able to make transfers in case of minor delays and the vehicle will just wait in locations where possible and convenient. Finally, there are situations where there is a trade-off between the access and egress distance and the total travel time. For example, a bus stop next to the actual point of origin of a journey might be serviced by a bus every hour while a metro station a kilometer away might bring you to your destination every five minutes, in half the time; in extreme cases the nearest transit facilities may not even take your destination at all. All-in-all this means that public transportation users take into account a lot more variables that affect mode choice for public transportation including departure time, number of transfers Raveau et al. (2014), aspects such as reliability Cats and Gkioulou (2017), transfers to high or low frequency services Bovy and Hoogendoorn-Lanser (2005). In this chapter we consider the following factors:

- total travel time from door to door.
- access time to first transit stop, which can be bicycling, walk or a car ride to a Park+Ride location.
- egress time to the last transit stop, which can be bicycling, walk or car, and is not necessarily the same modality used to access the public transit network.
- number of transfers required.
- average adaption time: the average time a passenger needs to wait to catch the next shortest journey to the destination.
- perceived service frequency.

Specific to the Dutch context we also include:

- total distance traveled on the transit network, as the fare is based on distance.
- fraction of distance traveled on the train network, due to a lack of fare integration.

The results of this research are aimed to be fed into an *activity based travel demand model*. Such models predict the number of trips by generating a daily plan for a set of

synthetic individuals. They include underlying choice models for activity type and duration, activity location, travel departure time, transportation mode and sometimes other choices, such as coordination with other persons for activities or travel. In the prediction stage, travel times need to be estimated for several options for the daily schedule being constructed: this is often based *traffic analysis zones* (TAZ) in order to limit the computational effort. In many cases a skim matrix is computed. Such a matrix provides *point estimates* for the inter-zonal travel time. We propose a method that is still TAZ based, but takes travel time variability into account. This provides more information for each skim matrix cell.

1.2. BACKGROUND

Computing a skim matrix for many traffic zone pairs is computationally very heavy, as for each sampled point in each zone a complete "shortest path" tree has to be constructed from that point to all other points. Due to this computational complexity, initially this problem has been addressed by creating static models of the transportation network, simplifying the network from a complex time-dependent network to a simple network with static costs between each transit stop. Studies such as Blanchard and Waddell (2017), who published their work as an open source Python library, transform the timetable at a specific time-of-day and day-of-week into a simplified graph. Each vertex is a transit stop and each edge is the in-vehicle time derived from the timetable. Modeling the transit network does not model dwell times, boarding and alighting times and traffic congestion nor transfers. The pedestrian network consists of three components: pedestrian access to and egress from the transit network and pedestrian-pedestrian for uni-modal pedestrian trips. To model the access to transit and egress from transit, each transit stop is connected to the nearest pedestrian node. As not every transit stop is serviced every minute, waiting time is modeled with the assumption that passengers arrive at transit stops randomly, this way, the average waiting time can be formulated as the average interval between each departure. Other studies using similar representations are O'Sullivan et al. (2000), Curtis and Scheurer (2010), Delmelle and Casas (2012), Tribby and Zandbergen (2012). This static representation is limited as it does not model the concept of transfers. The number of transfers is important in two ways, it affects the travel time and the reliability of the journey and is an important characteristic that influences mode choice. Hadas and Ranjitkar (2012) propose an extension where the number of required transfers is modeled along with the nature of each transfer: e.g. transfers that do not involve walking, street crossing transfers, same sidewalk transfers and direct trips. In their variants, the timetable is still statically defined but in a more detailed manner than just between pairs of stops.

With the increasing availability of data, computational power and improved algorithms, the focus has shifted to even more detailed representations of transit accessibility. Work by Benenson et al. (2011), Salonen and Toivonen (2013), Tenkanen et al. (2016) use transit timetables to give better approximations of total journey duration at a given moment of time and day based on earliest arrival. However, these methods are still limited and do not capture two other aspects important to transit users: the frequency of connections and the wait times involved. In their study Tenkanen et al. (2016) calcu-

late all optimal journeys from all transit stops to all other transit stops in the Helsinki study area. Kujala et al. (2018) uses the transit routing algorithms by Delling et al. (2014) to perform a range query of Pareto-optimal journeys. This list of Pareto-optimal journeys describe all journeys departing from a given time period. The simplest version just optimizes on travel time with the tuple ($T_{departure}$, $T_{arrival}$), where for each advice applies that for each advice 1 and 2 in the list applies that ($T1_{arrival} < T2_{arrival}$ or that $T2_{departure} > T1_{departure}$ allowing for a later departure. A range query will look at all possible departure possibilities within the given time range, excluding departure possibilities where passengers will just have to unnecessarily wait at transfer locations. Typically the number of transfers is also included as a property, allowing for longer travel time as long as it reduces the number of transfers, as an additional transfer might only reduce the travel time by a couple of minutes. For most public transit passengers in the Netherlands this list forms their choice set for transit route choice, as many will use the route suggestions from their favorite public transport app. Kujala et al. (2018) use the Connection Scan Algorithm (CSA) by Dibbelt et al. (2018), a very simple algorithm that puts all possible connections between transit stops in an ascending departure time order in a single array, to optimize cache locality. This might be counter intuitive as it increases the number of operations, however due to faster access to data stored in the CPU cache the algorithm performs very well or better than algorithms that access the memory locations randomly. Another algorithm that performs well for computing a range of Pareto-optimal journeys is the Round-Based Public Transit Optimized Router (RAPTOR) algorithm by Delling et al. (2014). RAPTOR is a dynamic programming algorithm that is organized into rounds, where in each round n the algorithm finds the earliest arrival time for stops reachable in n transfers. By using the transit lines reachable from a transfer in round $n-1$ or those accessible from the origin, the algorithm runs until no better arrival time is found at any stop. At that point a shortest path tree is complete, with the additional bonus of journeys with a longer travel time but a lower number of transfers.

With this full list of Pareto-optimal journeys it is possible to define accessibility beyond just travel times. It is also possible to compute variables such as the service frequency by the number of Pareto-optimal journeys within the time range and the average adaption time metric, using the rooftop method defined in Guis and Nijënstein (2015). We take all the Pareto-optimal connections between Origin and Destination found using the range query, calculate for each minute in the time range the required adaption time that a passenger has to wait for the next best connection and average the adaption time per minute. This forms a better metric for the travel time that a passenger has to experience including waiting time before and after the journey. The higher the frequency, the higher the probability a journey will fit well to their schedule and thus reduce the time it will take between origin and destination.

1.3. CASE STUDY

To calculate Pareto-optimal journeys, we used an existing extension to MATSim (Horni et al., 2016) developed by the Swiss railways(Bundesbahnen, 2020). This library uses the RAPTOR algorithm from Delling et al. (2014)and the MATSim framework for transit schedules to calculate all Pareto-optimal journeys departing in a given time range from

and to zones. The library samples origin and destination locations from each zone, to get a realistic view of each zone, as the travel time within each zone has a variance depending on the proximity of the origin/destination to the nearest public transport stop. In this case study, we opted for the zoning system defined by Statistics Netherlands (CBS) at the smallest level of *buurt* or *neighborhood* level CBS (2019). This is the highest resolution level of detail with a significant number of (openly) available statistics such as demographics and income. Descriptive statistics on the size of these zones are listed in Table 1.1, both for the zones exclusively in the municipality of Amsterdam and all selected zones for this chapter. We limited the study area to the area of the '*Vervoerregio Amsterdam*', the transportation authority in Amsterdam and surrounding municipalities in the greater metropolitan region. For the transit schedules, we used the open data General Transit Feed Specification feed published for the Netherlands (OVapi, 2020). To reduce computational time we removed all services not relevant for our study area; specifically we removed all services not traversing the metropolitan region of Amsterdam. We selected the day with the most services as our study day, a Friday with night bus and train services and we selected a 2 hour time range in the morning rush hour between 07:30 and 09:30 for our range queries. We selected this time period specifically to capture more than one hour to capture hourly frequencies and the specific time period to capture additional bus services in the morning rush hour.

Table 1.1: Descriptive statistics of area size in used zones: Statistics Netherlands Buurt level. In this chapter we used the zones of the Amsterdam Transportation authority.

		Area size (km^2)			
	N	Min	Average	Median	Max
City of Amsterdam	478	0.015	0.411	0.215	10.121
Amsterdam Transportation authority	802	0.015	1.004	0.340	26.078

In this case study we made a number of modifications to the MATSim library. Firstly, to get a realistic distribution of origin and destinations, we extended the library to load a list of addresses from the *Dutch building registry* in order to sample buildings weighted based on their function and usable surface. This way, we get a good sample of both residential locations and prominent locations such as hospitals, schools and retail locations. Secondly, we modified the library to write all results between each sampled origin and destination point to disk, so we can build up a high number of observations for each zone pair.

To get a sufficiently high number of samples we ran the computation on the Dutch research compute cluster LISA, where running the query with 20 sample points in 802 zones took approximately 3.5 hours on a 16 core CPU node, including the time to write 1 gigabyte of gzip compressed CSV with the intermediate results to disk. Running the query with five sampled points in each zone took 25 minutes for each run and with ten sampled points took 67 minutes.
With these computational times in mind, we conducted 100 runs with five points in each

zone and hundred runs with ten points in each zone, to get a good number of runs. To look at the effects of a very high number of points per zone, we conducted hundred runs of 500 points per zone. Due to the quadratic growth of computational complexity and data output, we limited the 500 points per zone run to a group of 16 zones. Additionally to look at the difference between public transport accessibility and car accessibility we used the same library to calculate skim matrices to provide us with comparable data on car accessibility. Finally we calculated a single run to produce a skim matrix based on 75 points sampled in each zone. This took approximately 44 hours to produce on a machine with two Intel Xeon(R) CPU E5-2620 (32 threads in total).

1.4. RESULTS

Figure 1.1 visualizes the travel times from a single origin zone, calculated in a single run with 75 nodes sampled in each zone. As we are more interested in the travel times to popular places such as hospitals and schools we applied sampling with replacement weighted based on surface and building function, dividing the surface of residential addresses by a factor of eight. This way, when selecting 75 points, only 58 unique coordinates function as the origin location in Figure 1.1. With the stored intermediate results

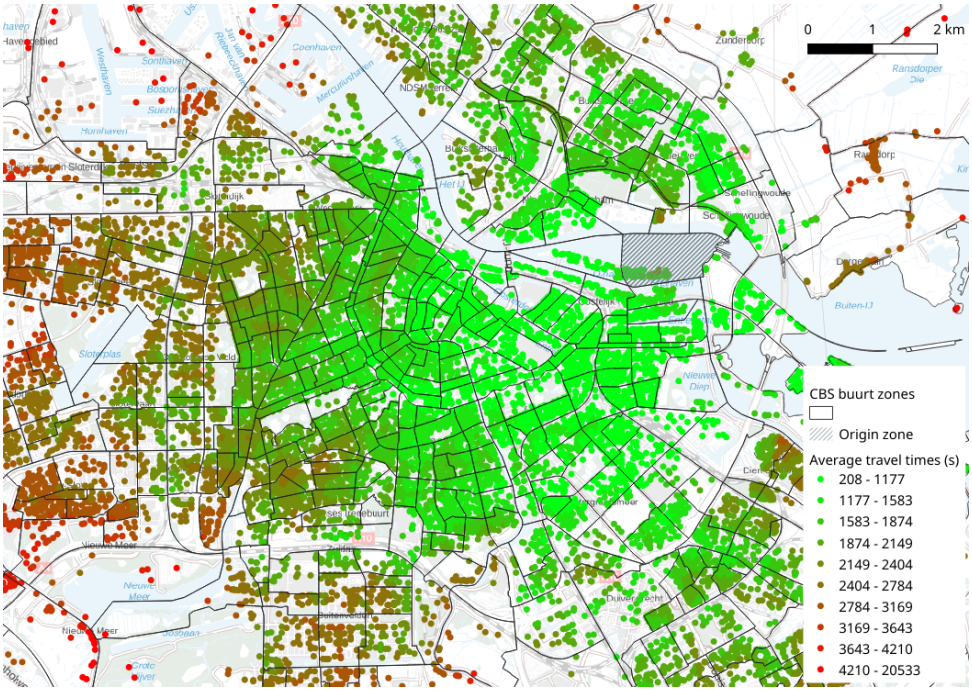


Figure 1.1: Travel times originating from KNSM-eiland neighborhood (indicated on map) based on sample size of 75 points per zone. Travel times for each destination are based on the average travel time from 58 origin points on KNSM-eiland.

for each single origin-destination pair between zones, we analyzed the data that forms

the average travel times in the skim matrices. This is a large collection of travel times as we have run the same query 100 times each and for each run we calculate $N * N$ travel times for the N points per zone. For an initial data exploration we looked at the histograms of travel times of a series with random origin-destination pairs, of which some are included in Figure 1.2. In this chapter we only included the histograms for the 100 runs with ten sampled points in each zone, as the hundred runs with five sample points in each zone had identical shapes.

What we discovered in the histograms is that the travel times for a large number of origin-destination pairs are bi-modally distributed and that only a very small number of origin-destination pairs are normally distributed. For our smaller focus group of sixteen selected zones, we did not spot any differences with regard to distribution when comparing the 100 runs of five, ten and 500 points per zone, beyond missing a few outliers in the runs with the smaller number of points. We included one example in Figure 1.3.

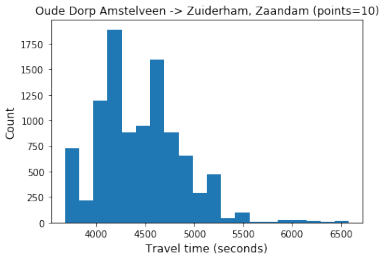
1.5. DISCUSSION

We see three main factors that reduce the number of normally distributed travel times, which play a role in both car and public transit but in different ways and intensity, *uniformity of travel speed*, *weighting of points per zone* and *number of points per zone*. We discuss each of these in turn below.

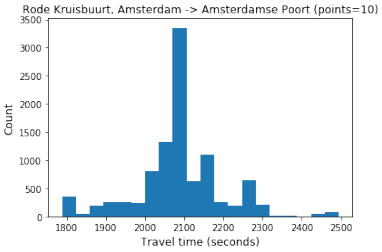
1.5.1. UNIFORMITY OF TRAVEL SPEED

Travel speed with car across the neighborhood will be uniform as long as all roads are accessible by car and one can drive directly from door-to-door and park there. If both origin and destination zone are structured in this way, then the distance would theoretically be ranged between $Length_{shortestPath}$ and $Length_{shortestPath} + Diameter_{origin} + Diameter_{destination}$, where $Length_{shortestPath}$ is the length of the path with the shortest distance between any two points in the respective zones. Thus the total distance cannot be lower than the shortest path between any point in each zone and cannot be higher than the sum of the shortest path with the diameter of the origin zone and destination zone. Assuming a single modality with a low variance in speed, this means that travel times are bounded by the same range. On the other hand public transit is by definition multi-modal with walk, bicycle or car as access and egress modes. And even if the origin and destination are the transit stop itself, there will be a high variability in travel speed, depending on mode, dwell times and the number of transfers; furthermore, depending on how well connecting services are timed there will be waiting involved as well. Transit times in the worst case would involve traversing the diameter of the zone with walk speed and in the best case there could be a high-speed train directly between origin and destination.

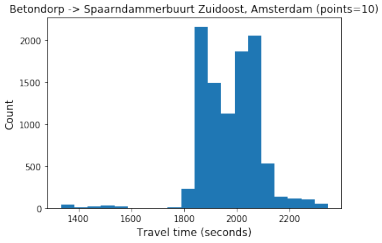
This effect has consequences however for studies focusing on infrastructure changes such as new metro stations, comparing the accessibility before and after; looking only at the average travel duration (i.e. a single number estimate) may give the wrong picture. An example of this effect is given in Figure 1.4, where a 700 meters distance between two origin points can increase the travel time to the destination zone by a factor of 2 in some



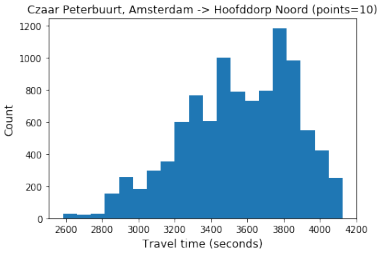
(a)



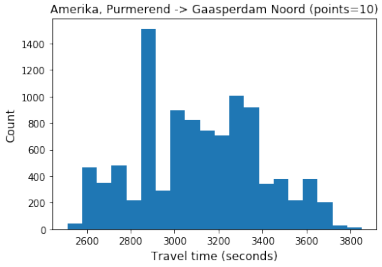
(b)



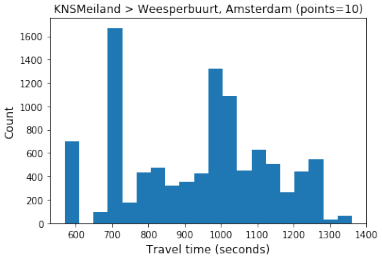
(c)



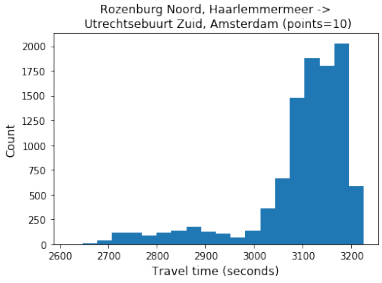
(d)



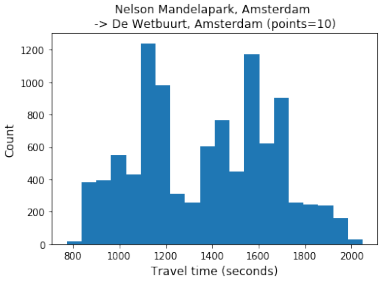
(e)



(f)



(g)



(h)

Figure 1.2: Histograms of travel times door-to-door using public transit between eight pairs of origin and destination zones in and around Amsterdam. Travel times were calculated 100 times between ten sampled points in the origin zone and ten sampled points in the the destination zone.

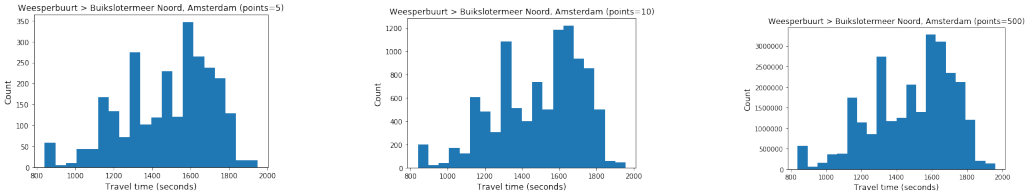


Figure 1.3: Histograms of travel times door to door using public transit for *(a)* five *(b)* ten *(c)* 500 points in both origin and destination zone.

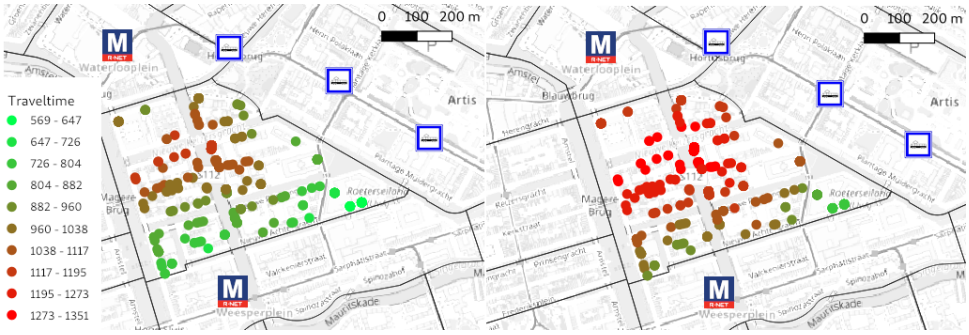


Figure 1.4: Illustrating the effect of shifting the origin position in a neighborhood 700 meter. The left figure originates from the east of the KNSM-eiland neighborhood with a tram service directly to the indicated tram stops. The right figure originates 700 meters further away. The bus line connecting with the metro networks passes both points.

parts of that zone. This is because one point has a direct tram service with limited walking whereas the other point is 700 meter from that tram stop. However the difference near the metro stations is smaller as the metro is connected by a bus that passes both points. We see this as the cause of the bi-modality in Figure 1.2f, which describes this zone pair.

Additionally important to note is that in this chapter we used straight line distances to model the access and egress to transit stops; we expect that when using actual distances instead the variability of transit travel times would further increase.

1.5.2. WEIGHING OF POINTS PER ZONE

What we saw with both transit and car accessibility is that heavily weighted points influence the distributions of the travel time. Since we aim to sample plausible origin and destination choices and not just addresses, we applied weights to addresses based on their function and surface. Namely, certain single addresses such as a hospital will function as the destination for thousands of trips per day whereas a single home is likely only the destination for a handful of people. Hence for the points we selected a single address will occur multiple times. This has an effect on the distributions as well, as the repetition of the same destination will increase the number of duplicate travel times. However with transit we see an additional interaction effect as those popular destinations will be made more accessible in well designed transit systems. For example in Figure 1.4 there is

a university building in the south of that neighborhood that is very well accessible and is selected multiple times in our sampling and forms a portion of that peak of 700 seconds.

1.5.3. NUMBER OF POINTS PER ZONE

What we noticed is that the number of points per zone has little influence on the distribution of the travel times. It might be better to repeat a lower amount of points per zone multiple times instead of running a higher amount of points.

1.6. CONCLUSION

Travel time level of service matrices for public transit containing a single value for each TAZ pair are commonly used. They represent point estimates applying to uni-modal travel. In this chapter, we introduced real multi-modal trips by accounting for first and last mile legs of journeys as well as for mode change and transfers. TAZ to TAZ travel times were computed by sampling locations in the origin and destination zone and by computing sets of Pareto-optimal travel options. We showed that in many cases the total travel time is not normally distributed but rather has a bi-modal or even tri-modal distribution. We also showed how this originates from accessibility properties of public transit stops. Elaborated experiments showed that the distribution for the travel time can be accurately estimated using ten or a few dozens of locations sampled in each TAZ. Careful sampling based on an address database specifying location types (residence, school, shop, hospital, etc) is required.

Extension of this research aims to replace the point estimates used in classical skim matrices by a specification of the travel time distribution. A possible research path is to discover typical kinds of distributions, classify them and estimate parameters to describe them. Each cell in the skim matrix will then consist of the distribution type identification and its parameters.

2

The Impact of Public Transit Frequencies on Door-to-Door Travel Times

Using modern transit routing algorithms it became feasible to compute all fastest travel options between all traffic zones in larger cities, allowing for an in-depth analysis of how frequencies affect journeys between different origins, destinations and departure times.

We use the rooftop method to calculate a realistic model of how a public transit user may perceive travel time, taking into account waiting time and/or adaption time to fit appointments in someone's schedule. The higher the frequencies, the lower those waiting times will be, and vice versa. The rooftop method calculates a travel impedance for any given moment in the travel time. Furthermore, as few journeys will start and end at a transit stop, some walk component is often also involved. We sample 6693 addresses for 799 zones to compute travel times door-to-door in Amsterdam and surrounding area, explicitly including walk access and egress time to and from transit.

In this chapter, we focus on the transit timetables before, during, and in the current phase of the COVID-19 pandemic in order to investigate the effect of changed schedules on accessibility and mobility by public transit. This is particularly relevant for services that have been reduced and may remain reduced for the near future moving ahead. We expect this application of methods outlined in this chapter to be of interest to public authorities and transit providers in making difficult decisions during COVID-19.

Koch, T., & Dugundji, E. R. (2021). Determining the effect of lower public transit frequencies in COVID-19 timetables on perceived door-to-door travel times using Pareto-optimal range queries. Proceedings of the 100th Annual meeting of the Transportation Research Board.

2.1. INTRODUCTION

As the COVID-19 pandemic caused abrupt and drastic changes in mobility demand and as some surveys indicate may result in lasting changes as working from home becomes more common. ANWB, a Dutch travelers association in the Netherlands, supporting all modes of travel including car, public transport, motor scooter, bicycle and e-bikes, found in a survey among a representative panel of 2000 members that around 60% would like to work from home two days or more. If such sentiment would become reality that would mean a drastic shift in mobility policy and reduce pressure especially on the peaks of traffic intensity. A drastic shift in mobility patterns would also mean that transit timetables have to be reworked to meet a new pattern of demands. For example, this could lead to a decrease of the frequencies in what are now the peak hours and perhaps increasing frequencies at different moments.

Various methods exist to measure the effect by changes on the timetable on accessibility of people to access jobs, hospitals or retail. The methods to calculate travel-times have become more and more sophisticated as better algorithms and increased computational capacity became available. Whereas initially transit accessibility was calculated using simple static networks delivered based on travel time between different stops, by 2012 it was possible using open source applications such as OpenTripPlanner (Byrd and Emory (2012)) to calculate all travel duration from a given location to any other location in the city based on a timetable based network.

However, these methods are still limited as they do not fully capture the time dependent nature of transit accessibility, as the transit schedule will never fully fit towards the schedules of the users. That means that people have to adapt their schedules to transit, departing slightly earlier from home to still arrive on time for your work appointment, or waiting at the train station to catch the first train home. To the users, this means that this wait time is part of how they perceive the travel time from door-to-door using public transport. In this chapter we will investigate how to compute this perceived travel time by calculating all possible optimal public transit journeys in a two hour time range, where optimal is the lowest duration between leaving from home and arriving at work or vice versa. Using this set of travel options we compute the perceived travel time based on the duration from door to door and the time in between each travel option. We first review the relevant literature on the computation of transit accessibility and perceived transit travel times. Next we will describe a case study based on the impact of COVID-19 and subsequent reduction of frequencies in the transit schedules, including a short timeline of how these changes arose. Subsequently we will show results using our methods on how the perceived transit travel times changed by the reduction in services.

2.2. BACKGROUND

In chapter 1 we calculated the average travel times for a single date. The main point of focus in that chapter was to study the effect of the sampling process to determine the points per zone that function as origin and destination, by running the calculation 100 times and storing the travel times between all sampled points in the statistical zones. In chapter 1 we noticed that travel time can vary depending on the exact point of origin and destination even within small zones of median 0.215 km^2 (0.39 square miles). De-

pending on the exact structure of the network, travel-times can differ in different ways depending on the interaction between the origin and destination. In comparison with travel time via bicycle and car we saw the effect of the heterogeneity of travel speed in transit, caused by three reasons. The first reason is that travel speed can differ for different modalities and can be stationary during transfers when waiting for the next vehicle. Secondly transit will almost always contain a walk segment towards and onwards from the transit network, the further away from a transit stop the higher the total travel time. Finally, since a sample of destination choices will include more popular destinations more than once and well designed transit service will prioritize more popular destinations, we saw an interaction effect of both selecting the same point multiple times and a lower travel time.

We hypothesized that these three ideas were the cause for the distributions we saw in the travel times, as many of the distributions were almost never normally distributed but often bi-modal. We concluded that when performing studies on the effect of something such as a new metro line, it would not be a valid comparison to make this comparison using skim matrices based on the travel-time averaged per origin-destination zones. In the present chapter we use the lessons from chapter 1 to make valid comparisons of accessibility in different timetables.

2.3. METHODS

To learn more about the comparisons in accessibility, we will review various methods on the aspect of calculating accessibility for a origin-destination skim matrix, look at existing methods to calculate continuously defined accessibility as time cubes. Furthermore, we look at routing algorithms that allow us to compute all shortest paths in a time range efficiently and a method from the literature to calculate a perceived travel time based on a series of departure/arrival times. We conclude with a survey of literature on differences in accessibility as a consequence of timetables changes or infrastructure changes.

2.3.1. SKIM MATRICES

Computing a skim matrix is computationally very heavy, as for each sampled point in each zone a complete "shortest path" tree has to be constructed from that point to all other points. Due to this computational complexity, initially this problem has been addressed by creating static models of the transportation network, simplifying the network from a complex time-dependent network to a very simple network with static costs between each transit stop.

Studies such as Blanchard and Waddell (2017) transform the timetable at a specific time-of-day and day-of-week into a simplified graph. Each vertex is a transit stop and each edge is the in-vehicle time derived from the timetable. This model does not include dwell times, boarding and alighting times and traffic congestion nor transfers.The pedestrian network consists out of three components: pedestrian access to and egress from the transit network and pedestrian-pedestrian for uni-modal pedestrian trips. To model the access to transit and egress from transit, each transit stop is connected to the nearest pedestrian node. As not every transit stop is serviced every minute, waiting time is modeled with the assumption that passengers arrive at transit stops randomly, this

way the average waiting time can be formulated as the average interval between each departure.

Other studies using similar representations are O'Sullivan et al. (2000), Curtis and Scheurer (2010), Delmelle and Casas (2012), Tribby and Zandbergen (2012). This static representation is limited as it does not model the concept of transfers. The number of transfers is important in two ways: it affects the travel time and the reliability of the journey and is an important characteristic that influences mode choice. Hadas and Ranjitkar (2012) proposes an extension where the number of required transfers is modelled along with the nature of each transfer: e.g. transfers that do not involve walking, street crossing transfers, same sidewalk transfers and direct trips.

Nassir et al. (2016) developed a utility-based travel impedance measure capable of capturing passenger behavior and their subjective perception of impedance when using public transport. Nassir et al. (2016) propose a measure that is time-dependent and estimates the realization of travel impedance for an origin-destination pair. They propose a nested logit 'logsum' over a generated set of plausible path options where a systematic utility is evaluated based on a discrete choice model that was previously developed and calibrated for transit passengers in the greater Brisbane, Australia area. Transit travel times are calculated from a fixed departure time at 7 am. They conclude that their measure incorporates all reasonable paths and mode alternatives and thus is able to correct for possible unobserved heterogeneity in their utility model and is able to capture the diversity benefit that a transit service can offer to a community of passengers.

2.3.2. CONTINUOUSLY-DEFINED ACCESSIBILITY

With the increasing availability of data, computational power and improved algorithms, the focus has shifted to even more detailed representations of transit accessibility. Work by Benenson et al. (2011), Salonen and Toivonen (2013), Tenkanen et al. (2016) use transit timetables to give better approximations of total journey duration at a given moment of time and day based on earliest arrival. However, these methods are still limited and do not capture two other aspects important to transit users: wait times and the frequency of connections as travel times vary over the day. A study with more focus on the temporal variability has been conducted in Anderson et al. (2013), in this study new methods are proposed to represent accessibility of a scheduled based transit system as a continuously-defined accessibility function (CDAF) of desired departure time, defined for all time points for each origin. They developed a CDAF function for four stops in the transit system in Minneapolis selected to show an example of a Urban Local, Major Transfer Point, Suburban Local and Suburban Express stop. Construction of the CDAF for each stop begins with identifying all trip departures in the time range of interest. For each selected departure the departure time and a vector T representing the travel times provided by that trip to all reachable destinations are kept. To store the number of opportunities at each reachable destination, a vector O is also established. With both these vectors it is possible to implement a wide range of different accessibility metrics.

In the study by Anderson et al. (2013) accessibility is calculated as for maximum travel time and the result defined as the number of destinations reachable within that threshold. This dramatically simplifies the calculation but does mean that every value of accessibility has to be reported separately for each different threshold and that their

model cannot be finely calibrated for varying user preferences and value of time. With this function, accessibility is calculated at each departure time for each possible trip. A time sampling interval is then selected and the calculation moves backwards through time from each trip departure time, applying the same function to the next trip's travel time vector subtracting the current time offset from each element. When the departure time for the previous trip is reached, the process is restarted using the travel time vector for that trip if the accessibility provided by the previous trip is greater than the accessibility provided by waiting for the next trip. This makes it possible to calculate accessibility A at each time point t, given by the CDAF in the following equation:

$$\begin{cases} A_{T_1} + \Delta_{t_1,O} & \text{if } t \leq d_1 \\ A_{T_2} + \Delta_{t_2,O} & \text{if } d_1 < t \leq d_2 \\ \dots & \\ A_{T_n} + \Delta_{t_n,O} & \text{if } d_{n-1} < t \leq d_n \end{cases}$$

Owen and Levinson (2015) use the work by Anderson et al. (2013) on continuous transit accessibility to look at departures at each minute to calculate the accessibility to jobs by transit. Owen and Levinson (2015) presents an accessibility-based model of aggregate commute mode share and focuses on the share of transit relative to auto.

2.3.3. GENERAL TRANSIT FEED SPECIFICATION

To compute the measure of travel time, schedules were used from publicly available schedule data in General Transit Feed Specification (GTFS) format. The calculation method used by Anderson et al. (2013) follows the method by Krizek et al. (2007) that uses proprietary software to calculate transit travel times. The results from this study confirm that accessibility is significantly overestimated by measuring a single point of time and show that trip frequency is more valuable for sustained accessibility than high accessibility on individual trips.

The study by Owen and Levinson (2015) is a study building on the CDAF function to provide an accessibility-based model of aggregate commute mode share focusing on transit relative to auto. Furthermore, they provide an estimation of a binomial logit model predicting the likelihood that a commuter will choose transit rather than auto for a commute trip based on aggregate characteristics of the surrounding area.

To provide the transit travel times instead of a proprietary library, they use the open source library OpenTripPlanner that operates on a graph representing road, pedestrian, and transit service. The transit travel times are calculated by constructing a shortest path for each origin, repeatedly every minute between 7 and 9 am as the departure time. For transit it uses census blocks as origin and destinations but the paper does not appear to describe how it is being accounted for spatial variability of transit in a census block zone. The paper concludes that continuous evaluation of accessibility as provided by transit systems are a promising metric to use in ridership and mode share modeling.

Tenkanen et al. (2016) uses a grid system of 13231 cells of 250x250 meter to calculate the travel times of public transport, bicycle, walk and car between each centroid of each cell. Travel times by public transportation have been optimized using 10 different departure times within the calculation hour using a so-called Golomb ruler that can be used to

gain maximal representation of departure times within one hour. The fastest route from these calculations is selected for the final travel time matrix.

2.3.4. PARETO-OPTIMAL RANGE QUERIES

A list of so called Pareto-optimal journeys describes all journeys departing from a given time period. The simplest version just optimizes on travel time with the tuple ($T_{departure}$, $T_{arrival}$), where for each advice applies that for each advice 1 and 2 in the list applies that $T1_{arrival} < T2_{arrival}$ or that $T2_{departure} > T1_{departure}$ allowing for a later departure. This is similar to the CDAF by Anderson et al. (2013) but with travel time as a accessibility function.

A range query will return all Pareto-optimal journeys departing or arriving in a given time range. For most public transit passengers in the Netherlands this is their choice set, as many will use the route suggestions from their favorite public transport app.

A range query will look at all possible departure possibilities within the given time range, excluding departure possibilities where passengers will just have to unnecessarily wait at transfer locations. Typically the number of transfers is also included as a property, allowing for longer travel time as long as it reduces the number of transfers, as an additional transfer might only reduce the travel time by a couple of minutes.

Kujala et al. (2018) use recent developments in transit routing algorithms to perform a range query of Pareto-optimal journeys between all public transport stops in the Helsinki region. To do so they use the Connection Scan Algorithm (CSA) by Dibbelt et al. (2018), a very simple algorithm that puts all possible connections between transit stops in an ascending departure time order in a single array, to optimize cache locality. This might be counter intuitive as it increases the number of operations, however due to faster access to data stored in the CPU cache the algorithm performs very well or better than algorithms that access the memory locations randomly.

Another algorithm that performs well for computing a range of Pareto-optimal journeys is the Round-Based Public Transit Optimized Router (RAPTOR) algorithm by Delling et al. (2014). RAPTOR is a dynamic programming algorithm that is organized into rounds, where in each round n the algorithm finds the earliest arrival time for stops reachable in n transfers. By using the transit lines reachable from a transfer in round $n-1$ or those accessible from the origin, the algorithm runs until no better arrival time is found at any stop. At that point a shortest path tree is complete, with the additional bonus of journeys with a longer travel time but a lower number of transfers. RAPTOR has a range query capability to efficiently compute bi-criteria range queries, outputting full Pareto-sets of journeys for all departures within a time range.

2.3.5. ROOFTOP METHOD

With this full list of Pareto-optimal journeys it is possible to define accessibility beyond just travel times. When increasing or decreasing frequencies of the transit network the effect may not be directly visible when looking at the travel time of a single journey, depending on how well connections still fit together. However, in the real world people are looking for mobility that fits their agenda. For example if a person wants to arrive at work by public transport in the morning by 9:00 a.m., this passenger will look for a jour-

ney that will arrive as late as possible, while leaving some room for possible delays. The optimal journey would consist of a journey that arrives just in time, but if frequencies are reduced the probability that such journey exists meaning that the perceived travel time of public transit will increase, even when the in-vehicle time stays the same. This is also shown in stated preference surveys among Dutch public transit users such as Schakenbos et al. (2016) where a headway of 30 minutes instead of 15 minutes will add seven to twelve minutes generalized travel time to the total utility of a trip.

To calculate this perceived travel time we used the "Rooftop" method described by Guis and Nijënstein (2015) and the Passenger Demand Forecasting Handbook by Rail Delivery Group (2009). This method calculates three aspects of the perceived travel time:

1. In-vehicle duration.
2. Transfer impedance: a value-of-time for each additional transfer; this value indicates the level of travel time gain necessary to make an additional transfer worthwhile.
3. Average adaption time: the duration between the arrival time and the ideal arrival time; this is the time necessary for a passenger to adapt the time table with their own agenda.

These three values can be calculated at any moment, as the average adaption time will differ for each moment. The three values combined are the travel impedance for that moment and the average travel impedance for a time period is called the level of service and a quantitative indication of the attractiveness of public transport on that origin-destination pair.

In the research reported in this chapter we will not apply a transfer resistance, but in the study by Guis and Nijënstein (2015) they based this resistance on work by Schakenbos et al. (2016) that estimated based on a static preference survey and a mixed logit model. This model estimated a transfer resistance of 5 minutes for transfers between trains and 20 minutes for transfers between train and bus/tram/metro. One of the reasons for a higher transfer-resistance between train and bus/tram/metro in the Netherlands is a lack of fare integration, as passengers will always be confronted with a 90 eurocent entrance rate.

This method is called rooftop as plotting the travel time over an hour gives a roof top pattern as shown in Figure 2.1. In this figure we see an O-D relation with a headway of 15 minutes: a departure opportunity at :00, :15, :30, :45. We have plotted the impedance for each service as a line with a different color. As the impedance of an additional minute will decrease gradually as the adaption time increases, this is modeled as a gradual reduction of the slope. The gray area plotted indicates the travel impedance faced by the user at any time and the average travel impedance is the indication for the level of service between the origin and destination.

- In Figure 2.1 we see that a person with a desired departure time coincides exactly with a departure opportunity will only be faced by the in-vehicle duration and a

potential transfer impedance. However, someone who arrives at the station at any other moment will also be faced by the adaption time, which will increase after a opportunity has left the station and will only decrease again once halfway between two opportunities, marking the roof-top.

- If we increase the frequencies of a service and improve the headway between services down to 10 minutes, as we did in Figure 2.2 we see that the rooftops have a lower height as the adaption time decrease and passengers are better able to find a suitable service without adapting their own agenda. This shows that increasing frequencies that reduces the headway between services improves the level of service.

- If we adjust the timetable such that the headway is not constant between services as we did in Figure 2.3 we see that some of the rooftops got lower (65) but the other two roof ended up much higher, as passengers arriving around the hour will be faced with higher wait times. On average the travel impedance increase and thus the level of service decreased.

- If we add services with an inferior travel time as we did in Figure 2.4, we see that the level of service is not affected as even at the depart time of the inferior options, the travel-time of these options is higher than the travel-time + adaption-time of the faster options.

2.3.6. CHANGES IN ACCESSIBILITY

The study by Grengs (2004) looks at measuring change in small-scale transit accessibility in terms of how many jobs are accessible, analyzing the transit routes serving one neighborhood each in Buffalo and Rochester, New York. The main question addressed in this study is firstly whether transit-dependent people with a low income living in inner-city neighborhoods lost accessibility by transit to jobs during the 1990s. Secondly, how much of that reduction in accessibility was caused by changes in transit server and not a spreading out of land use. In this study the analysis of transit travel times is limited to transit journeys without transfers as building a network to include transfers would increase the computation time exponentially and was considered to be out of scope of their study.

Farber and Fu (2017) compare how transit travel times and accessibility have changed in response to network and service modifications. To do so they proposed a novel data object called the public transit travel time cube, this cube contains the shortest path transit travel time between sets of origins and destinations in a city, at all times of day. This study looks at how travel times were impacted by service cuts and expansions in the two regions respectively and the impact this had on jobs accessibility. Furthermore, they use the travel time to compute to study the last mile problem and investigate the last mile problem and compute travel time savings and reliability improvements of the bicycle transit combination.

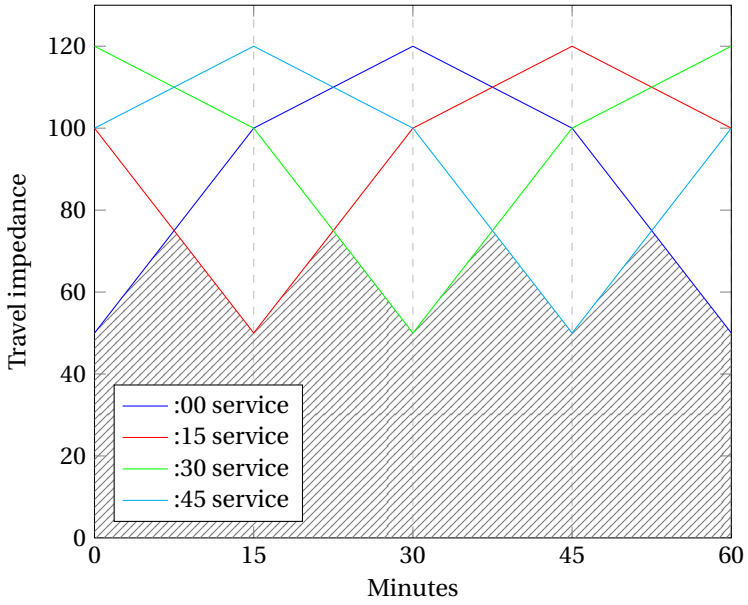


Figure 2.1: Plot of travel impedance when applying the rooftop method, each line showing the impedance of four departure opportunities each at the quarter of the hour. We see the rooftop effect of the adaption time peaking increasing after missing a departure opportunity and decreasing when halfway to the next opportunity. The horizontal axis indicates each minute of the hour and the vertical axis the travel impedance.

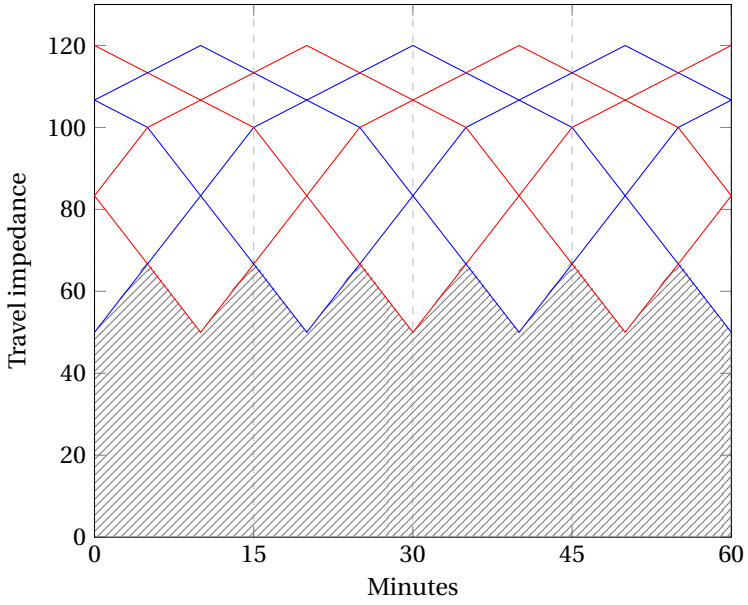


Figure 2.2: Again the rooftop method applied, but now with a higher frequency, each service with an alternating color. We see that the rooftops are now lower, indicating that the perceived travel impedance is lower and thus a higher level of service.

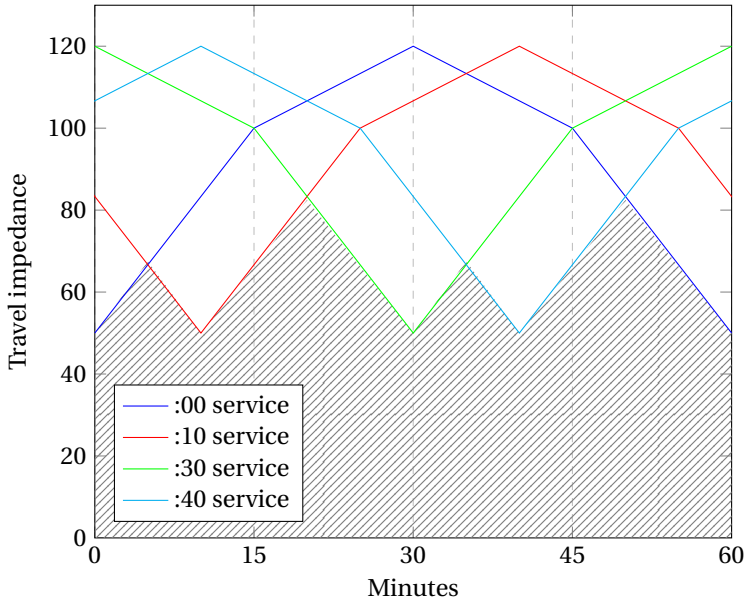


Figure 2.3: Plot of travel impedance when applying the rooftop method, each line showing the impedance of four departure opportunities each at the quarter of the hour. We see that if the departure opportunities are in an irregular pattern, the wait time between services varies. Over the hour the level of service decreases as the wait-time on average increases.

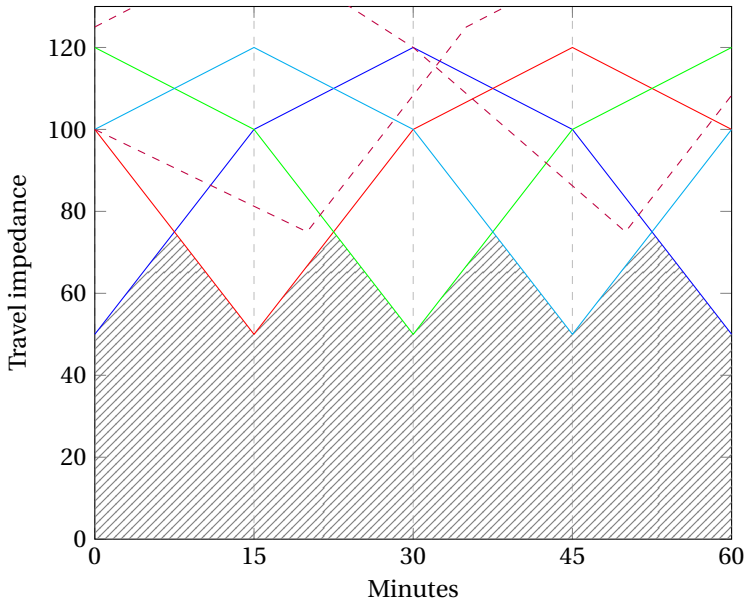


Figure 2.4: Plot of the rooftop model when we add two options with an inferior travel-time and/or transfer impedance. As the impedance of these two options in dashed purple is higher than the other four options, even when including the adaption time to those services the level of service is not affected

The public transit travel time cube is a three-dimensional array, $T = [t_i, j, m]$, of estimated transit travel times, with i and j the origin and destination zone respectively and m the computed for a minute in the day. This makes the dimension of the cube: $NxNxM$ for N zones and $M = 1440$ for the minutes in a day for a cube that contains the public transit travel time from all origins to all destinations at all times of day representing the latent structure of public transit connectivity in a region. In the implementation of Farber and Fu (2017) the travel time is calculated with a toolbox of a proprietary geographic information system (GIS) software. This software allows their users to calculate an origin-destination matrix with travel times at a specific departure time. These matrices indicate the travel time between the centroid of each zone and 0 for intrazonal trips as there is only a single point in each zone. To produce the cube they calculate such a matrix for every minute of the day and combine it into a cube.

Through three case studies, the study shows how travel time cubes can unlock new ways to describe the spatio-temporal patterns of travel times and use these in the development of more sophisticated measures of accessibility and quantify the impact of accommodating joint bicycle and transit trips. In the first case study they compare travel times over time-of-day, looking at how travel time has changed based on absolute travel times and a decomposed time series using the Fast Fourier Transform algorithm that produces a distribution of frequencies that combine to reproduce the actual time series, the second case study access to jobs using travel time cubes is being assessed and the third case study looks at the effects of bicycle use on the last mile problem.

2.4. CASE STUDY

We will describe our case study starting with a short timeline on how the timetables were affected by the pandemic. Next, we explain how we selected our study period and study area. Finally, we describe how we calculate all travel times in three two-hour periods between thousands of origin and destinations.

2.4.1. TIMELINE OF THE PANDEMIC AFFECTING PUBLIC TRANSIT IN THE NETHERLANDS

The first official case of COVID-19 in the Netherlands was diagnosed on the 27th of February 2020. By the 9th of March of 2020 there was a request by the authorities that people in the south province of Noord-Brabant to work as much as possible from home and on March 10th all events with over a thousand people were forbidden. As the pandemic was not contained to this province, these measurements were extended to the entirety of the country on March 12th and people were requested to work from home as much as possible. By March 13th all events with over a 100 people were forbidden, museum's, universities, theaters, and sport clubs were closed. On March 14th a sudden announcement closed all bars and restaurants at 6 pm that night and that after an initial resistance based on lower risk, the decision was made to also close K12 and high-schools.

Despite this extreme rapid shift in demand of mobility in this initial week only some small modifications were made to remove special peak train services. However, 7 days later, based on a high sick leave among staff members and the lack of passengers, the

decision was made by transit operators and authorities to reduce timetables in order to ensure continued service to essential workers. In effect this meant on the train network the service was reduced to twice an hour in every direction from each station and removing almost all Intercity trains leaving only regional trains stopping at every intermediate station. Bus, tram and metro service was generally reduced to the usual Sunday service with some added frequencies in the (early) morning. To avoid a mismatch in demand and supply especially in the context of allowing 1.5 meters (approximately five feet) distance between each passenger, a stringent request was made to all passengers to only travel for essential purposes. The latter was not legally enforced, but generally adhered, to especially in the initial phases. Furthermore, in Amsterdam a single metro line 51 was suspended leading to those passengers having to make an additional transfer. This period of reduced service was in effect for 5 weeks.

By April 21st the first re-openings were announced, as primary schools would open in smaller groups for half the time. In preparation for that and in response to a slight increase of demand, transit agencies increased the level of service again. The intercity train network was resumed on April 29th to ensure at least one intercity service per hour in each direction from each Intercity station and local bus, tram and metro operators also increased their frequencies. On the same day in Amsterdam the local operator GVB resumed the service of metro line 51 and increased the number of frequencies of a number of tram and bus lines. Two weeks later it would further increase the services, moving towards the level of service found each year in the summer holidays. By the 28th of May, GVB reported in a press release to be transporting about 20 to 25% of it's usual number of passengers.

On the 1st of June, the original frequencies were restored for the train network. For the bus,tram, metro network in most cases a return was made to the summer timetable, which has lower frequencies. to avoid crowded situations, operators required a non surgical mask to be worn on board of the vehicles. Without crush loads, this would mean a return of 40% of the original capacity, meaning that the stringent advice to only use public transport for essential travel was still in place. Meanwhile, on June 1st, bars and restaurants were allowed to open for outdoor service, high schools were allowed to let students in, primary schools open fully June 8th and universities were allowed to let students take tests on campus.

On July 1st, the authorities dropped the stringent advice to only travel by public transport for essential purposes. However, as of 1st of August 2020, the request to work from home if possible is still in place and tourism to Amsterdam is lower than normal as US citizens are still restricted from travel. This all has led to a significant reduction in demand for mobility with GVB currently transporting around 40% of the passengers in greater Amsterdam than it had in the same period last year. As public transport at these ridership levels is probably not cost effective and the end date of this situation was not known at that point, the transportation authority in Amsterdam gave permission to reduce the frequencies of the metro network from 6 to 5 per hour and reduce the frequencies on certain Bus Rapid Transport (BRT) lines.

2.4.2. STUDY PERIOD

We selected only Thursdays to study, as Thursdays are generally representative for all other business days while Thursday has commonly has the highest number of passengers.

- For our baseline we selected two different days, as the week before the lockdown there was noticeable interference from a works project that finished up in the first weeks of the lockdown. The days we selected as the baseline were February 13th and March 5th 2020.

- As it took some time for all the changes to get processed and end up correctly in the published timetables, we selected the date of Thursday April 24th as the day most reflective of the most heavily reduced timetable between March 21st and April 28. And to reflect the third period with slightly increased service we picked Thursday May 21st.

- As the final date we selected June 25th, measuring the as of writing current timetable. This current situation has a currently unknown end date, perhaps not changing before a vaccine or better therapy is found.

To measure accessibility during the day, we selected three different time ranges for each day, measuring the morning peak between 7 and 9 am, the evening peak between 4:30 and 6:30 pm and off-peak between 11:30 am and 1:30 pm.

2.4.3. STUDY AREA

We limited our study area to the region of transport authority for greater Amsterdam (in Dutch: Vervoerregio Amsterdam), which includes the municipality of Amsterdam and fourteen neighbouring municipalities: Aalsmeer, Amstelveen, Amsterdam, Beemster, Diemen, Edam-Volendam, Haarlemmermeer, Landsmeer, Oostzaan, Ouder-Amstel, Purmerend, Uithoorn, Waterland, Wormerland and Zaanstad. The reason to limit it to this smaller area was the exponentially increasing computational time involved with scaling further spatially. Instead, we preferred to keep the number of sample points high and look at effects in this specific context with a still interesting mix of urban and suburban context.

2.4.4. SAMPLING OF ORIGIN AND DESTINATION PAIRS

As measuring from all addresses would be computationally prohibitive, we used a sampler to select a smaller set of addresses. To make this selection of addresses more reflective of destination choices, we weighted based on the purpose of that address (such as residential, education, retail, commerce, etc.) and the surface-area of the building. To reflect real destination choice, we allowed the same address to be sampled multiple times.

As we saw interesting effect of the exact origin and destinations selected on the determination of door-to-door travel time with public transport in our study Koch et al. (2020), we slightly adjusted the sampler from that paper to allow for more points to be sampled to have more probes of the spatial variability. Our data source to sample is the national registry of buildings and addresses, containing information such as location, surface area and the generic indication of purpose of that building such as education, residential, commercial, retail, etc. The surface area is the sum of the surface areas of all floor levels in the building.

As our zoning system we use the Dutch national bureau of statics zoning system at the smallest public level of the neighborhood (in Dutch: buurt). These zones in Amsterdam have a surface area of median 0.411 km^2 (0.16 square miles). Instead of a low number of points per zone we opted for a high number of points with the constraints that any point within a zone cannot be closer than 200 meters to an already included address. This results in at least 1 and at most 64 points per zone, on average 8.3 points per zone and median 6 points per zone and in total 6693 points. To remove any effect caused by different sets of addresses that we saw in Koch et al. (2020), we used the same set of addresses in every measurement.

2.4.5. CALCULATION OF TRAVEL TIMES

To establish perceived travel time we needed to calculate all optimal travel options in a given time range. Based on our earlier experience with transit routing algorithms, we decided on the Round-Based Public Transit Optimized Router (RAPTOR) algorithm from Delling et al. (2014) as the algorithm to compute. As many open source implementations exist of RAPTOR, we opted to use an existing extension to MATSim (Horni et al., 2016) developed by the Swiss railways (Bundesbahnen, 2020). This library used the MATSim framework for transit schedules to run RAPTOR to calculate all Pareto-optimal journeys departing in a given time range from and to zones.

We modified this library to output all intermediate results leading to a large dataset with 534,188,098 records describing the descriptive statistics for three time periods for four days between 6693 origin-destination pairs. With this data set it is possible to explore accessibility differences for four days during and after the COVID-19 crisis.

2.4.6. WEIGHTED AVERAGE PER ZONE

As this high number of records made it difficult to visualize the results in an interactive manner, we opted to calculate a weighted average of travel-times per zone. To do so, we took the weight of each address and added this weight to the nearest address that was part of our sampled set of origin and destinations. In this way, the weighted average is a good representation of travel-time based on the relevancy of address while trying to limit the effect of the spatial distribution of travel times across each zone.

2.4.7. INTERACTIVE TOOL

To visualize travel times and changes in accessibility interactively we built a simple web application that lets any user click on a zone in order to render results for all destination zones. The application allows the user to select the date and time period to visualize the travel-times and differences in travel-time between each date and time combination.

2.5. RESULTS

In this section we present four views of our results: two approaches to considering travel time, and accordingly, two approaches to evaluating changes in accessibility.

2.5.1. TWO APPROACHES TO TRAVEL TIME

Running the computation is relatively quick, and we are able to compute 3 2-hour periods for the entire city of Amsterdam in a bit less than 2 hours on a 16 core machine. This could have been less if we did not opt for writing all travel times between the 6693 selected points, as this meant writing 8 gigabytes compressed text to disk. These files did allow us to look at accessibility at point level as we visualized in Figures 2.5 and 2.6. In Figure 2.5 we visualize the perceived travel time including wait times to the Vrije Universiteit campus and in Figure 2.6 we visualize the median of all travel times found between 7 and 9 am, indicating the travel time without wait component.

2.5.2. TWO APPROACHES TO CHANGES IN ACCESSIBILITY

Looking at the effects on travel times of the different iterations of the timetables during the Corona pandemic in March to June 2020 we see that as expected the travel times increased between most origin and destinations. Specifically, we looked at two indicators of accessibility, the travel-time as based on the fastest option in the two hour range and travel-time as computed by the rooftop method to get an indication of frequency and travel-time.

For some origin-destination pairs we saw some interesting effects where both indicators diverged. We included one such example in Figure 2.7 with the fastest travel-time and Figure 2.8 with the travel-time based on the roof-top method, comparing the changes in accessibility from Amstelstation in the east of Amsterdam between Thursday March 6th and June 25th 2020. The changes in service between these two dates cut the metro frequency from five to four departures per hour on metro line 50, 51, 53, 54, with service unchanged for the North/South metro line 52 that still has a headway of six minutes. The bus schedules were mostly unchanged outside the removal of peak bus routes and reducing the frequencies on the Bus Rapid Transport network from the South of Amsterdam to Schiphol airport.

Looking at Figure 2.7 where we look at only a single travel, we see no changes in accessibility around the metro network, as all those stops are reachable without transfer or on metro line 52 with unchanged frequency. However, as the metro frequencies were reduced, a passenger arriving at an arbitrary moment probably will have to wait a few minutes longer than normal. This reduction in the level of service is noticed when applying the rooftop-method in Figure 2.8, where we see a degraded service along the western edge of the metro network.

2.6. DISCUSSION

We specifically picked the Vrije Universiteit (VU) campus as the destination to describe our case study for two reasons: the location of the university itself, and the medical center facility associated with it (VUmc).

The university itself is a major source of trip generation related to the 26593 bachelor

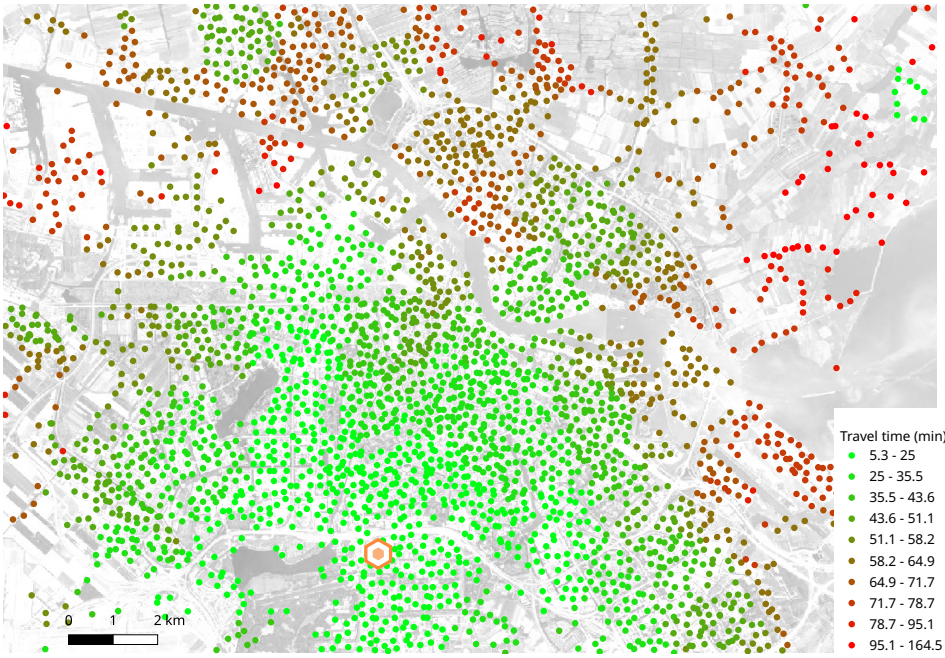


Figure 2.5: Travel times towards the Vrije Universiteit campus in the period departing between 7:00-9:00 a.m. on the 13th of February 2020. Each dot represents the *perceived travel travel time* including adaption time and/or waiting time. Green indicates low travel time and red indicates high travel time. For clarity all points with negligible travel time are omitted.

and master students that travel weekly to the campus from the region for their studies, with only a relatively small number of students living nearby the campus in Amstelveen. For these students, the campus serves as a hub for trip chaining to other activities. It is not simply a matter of students coming to campus from their residential location and returning directly, but instead often involving trip chain behavior to other locations. The campus thus is ideally accessible to and from a vast range of residential locations spread throughout the region, but also to and from other activity locations. This huge daily flux of students creates challenges for transit providers in providing sufficient capacity via sufficient frequency of service at peak periods. It also offers opportunities for the university board to consider the impact of changing lecture schedules and giving online options for course work.

In addition, the campus with the teaching hospital forms an important destination for essential workers keeping the hospital running, such as janitorial staff, nursing aids, nurses and doctors still working on site 24/7. In this regard, the accessibility is relevant not only spatially for the essential workers living at residential locations spread throughout the region, across all income levels, but also it is necessary to have 'temporal' accessibility throughout all hours of the day and night.

For public authorities, the impact of transit scheduling by transit providers is a care-

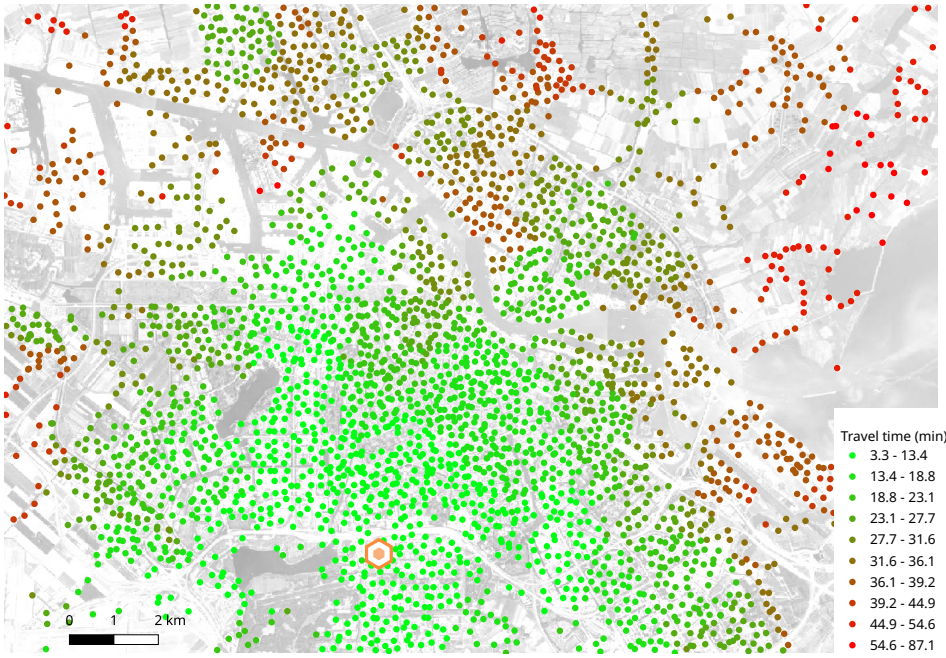


Figure 2.6: Travel times towards the Vrije Universiteit campus in the period departing between 7:00-9:00 a.m. on the 13th of February 2020. Each dot represents the *median travel duration* based on each optimal departure opportunity found in the two-hour period. Green indicates low travel time and red indicates high travel time. For clarity all points with negligible difference are omitted.

ful balance between equity and finance. On the one hand, there is a public responsibility to provide equitable service for different segments of society. On the other hand, public transit is a semi-public operation requiring a sufficient level of service to be affordable, financed partly by revenue from operations and from public funds. The research methodology outlined here can support public authorities and transit providers in understanding the impact of scheduling from the perspective of the travelers in the system at a fine-grained spatial level, for all locations 24/7 across the entire transportation region.

2.7. CONCLUSION

When looking at how changes affect travel time as experienced by the population as we did for the service reductions for COVID-19, it becomes clear that there can be a gap between the travel time for a single journey and the travel time for a passenger who has to be somewhere at a given time.

Especially now, when many transit network operators are thinking about a reduction of frequencies as a response to the lower demand in mobility during this pandemic, it is important to keep in mind that accessibility of transit will vary for every minute of the day and that the reduction of frequencies may not be visible in analysis that ignores the

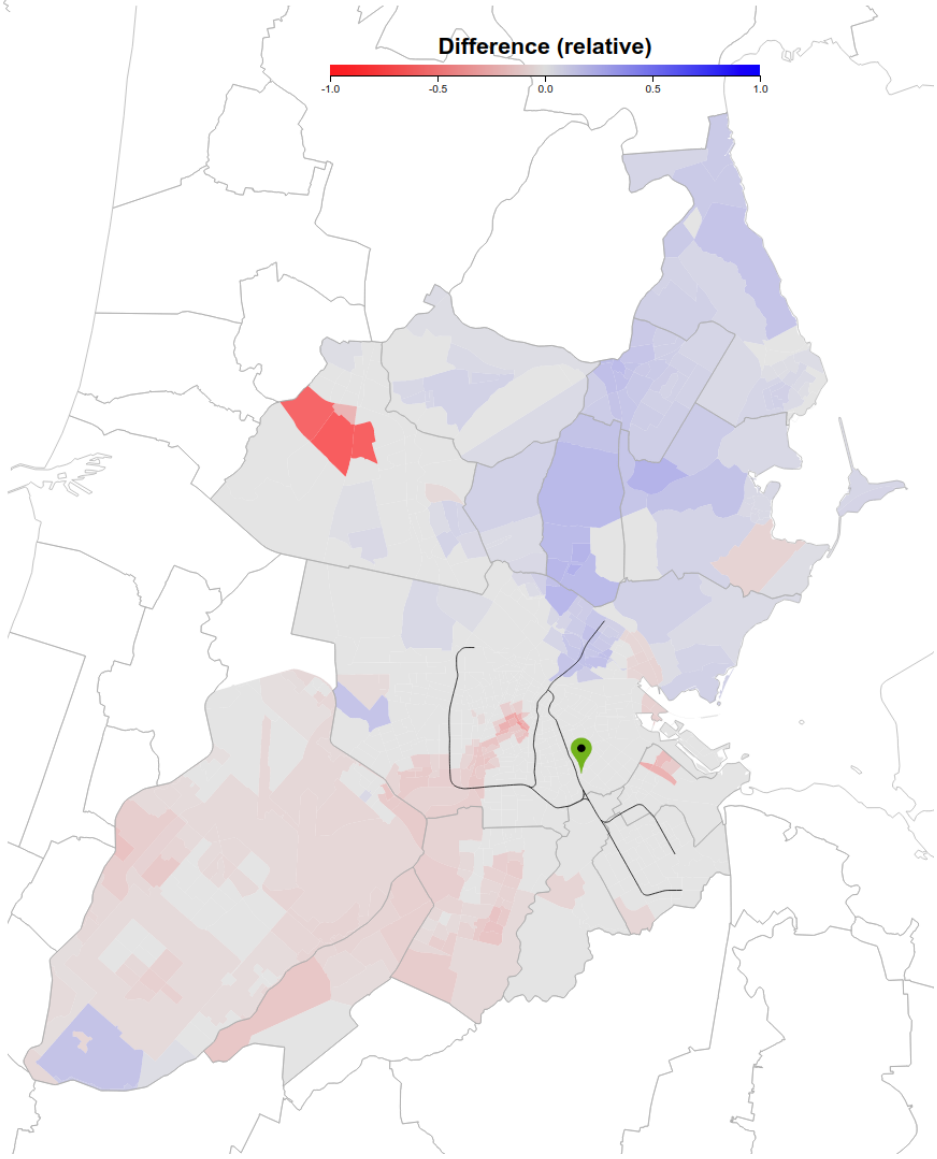


Figure 2.7: Difference in accessibility from Amstelstation (green marker) between March 6th 2020 and June 25th 2020 as measured by the shortest travel-time door-to-door in the early morning commute period (7 to 9 am) by taking the shortest travel time of any trip that departures between 7 and 9 am. A red shade indicates a longer travel-time, a blue shade indicates an improved travel-time and gray indicates a difference of less than 90 seconds. The black lines indicate the metro network of Amsterdam.

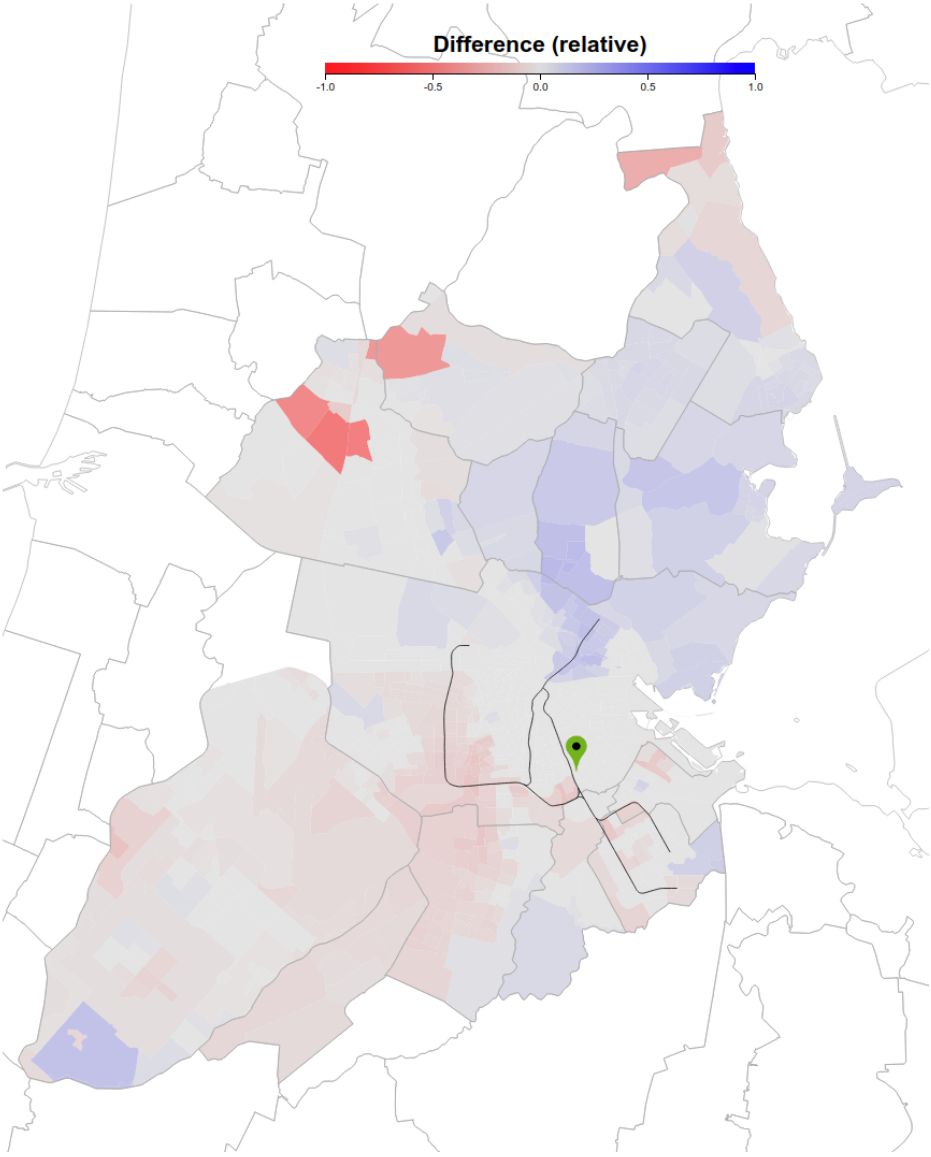


Figure 2.8: Difference in accessibility from Amstelstation (green marker) between March 6th 2020 and June 25th 2020 as measured by the perceived travel-time door-to-door in the early morning commute period (7 to 9 am) with the roof-top method. A red shade indicates a longer travel-time, a blue shade indicates an improved travel-time and gray indicates a difference of less than 90 seconds. The black lines indicate the metro network of Amsterdam.

temporal variance caused by the frequency component of each transit schedule.

The main suggestion we see to get the full potential out of this tool, would be to make these measurements of perceived travel-time more accessible to end users such as public policy makers and transit operators to further optimize transit schedules and understand how frequency increases or decreases would affect different demographics.

For future study we would like to compare perceived travel time with more accurate comparisons of perceived travel times by car and bicycle.

To make a fair comparison with bicycle and car it would only be fair when measuring the perceived travel time with that modality ALSO as door-to-door. This would require assuming a walk to the nearest available car parking spot on-street or off-street in a garage, or to the nearest available bicycle parking rack on-street or off-street in a bicycle parking facility. Ideally you would also include some cruising to find a parking spot and some form of traffic information to include congestion information.

Another very interesting aspect of study, especially in Amsterdam, would be to study perceived travel time of the "bicycle + public transit" combination. As travel by bicycle can increase the accessibility to and from higher frequency transit services such as metro, we hypothesize this would reduce the likelihood of having to wait for the next possible departure opportunity at a lower frequency transit stop such as bus.

3

SHORT-TERM FORECASTING OF OFF-STREET PARKING OCCUPANCY

Information and Communication Technologies have opened the way to guide recent developments in the field of parking. In this chapter, these technologies are applied to model a Decision Support System that gives insight into six months ahead parking occupancy forecasts for 57 off-street parking locations in Amsterdam. An effect analysis has been conducted into the influence of weather, event, parking tariff and public transport attributes on the parking occupancy. The most influential factors are the event variables, thunderstorms, average wind speed, temperature, precipitation, sunshine, and the addition of a public transport line. Parking tariffs did not significantly contribute to the model performance, which can be caused by the lack of data and time variability in the parking tariffs of the examined parking locations. The forecasting algorithms compared are Seasonal Naive Model as a benchmark approach, Box-Jenkins Seasonal Autoregressive Integrated Moving Average with and without exogenous regressors (SARIMAX and SARIMA, respectively), exponential smoothing models, and Long Short-Term Memory neural network. The SARIMAX model outperforms the other algorithms for the six months ahead forecasts according to the lowest Root Mean Squared Error (RMSE). By including the event factor, the model improved with 24% based on the RMSE. Weather variables improve the predictive performance with 8%.

Fokker, E. S., Koch, T., van Leeuwen, M., & Dugundji, E. R. (2021). Short-term forecasting of off-street parking occupancy. Transportation Research Record.

3.1. INTRODUCTION

The development of low-cost, on-demand shared mobility systems and the rise of the autonomous vehicle have made transportation the center of today's debate and discussion (Vazifeh et al. (2018)). Yet, innovations and developments in the field of parking are happening more quietly on various fronts, such as technology, research and design (Rosenblum et al. (2020)). One of these developments is the *reduction or elimination of parking minimums*, which is the minimum amount of parking spaces required. Cities at the forefront of parking progress reduce or remove parking minimums and replace them with maximums, eventually resulting in a decrease in parking supply (Kodransky and Hermann (2010)).
Another development is *dynamic pricing*. In dynamic pricing, parking prices vary based on dynamic factors. One factor is pricing based on the expected parking occupancy. Transport experts have found an optimal amount of 15% of unoccupied parking spaces to minimize the time spent "cruising for parking", which adds substantially to the severity of downtown congestion (Arnott and Inci (2006) and Barth and Boriboonsomsin (2007)). To reach this optimum, drivers need to pay a higher price to park in more desirable parking locations. Another factor is pricing based on the emission level of vehicles. In some European cities such as Amsterdam and London, cleaner vehicles pay a discount rate on parking charges, whereas a higher rate applies for vehicles that pollute more (Kodransky and Hermann (2010)). Nowadays, these developments have become even more relevant, as more public space is required by the COVID-19 pandemic. Major cities such as Athens, Paris and Berlin have set the goal to liberate public space from cars and make place for clean traffic, such as walking and cycling.
Just as many other affluent cities, Amsterdam is actively engaged in these developments in order to reduce the negative impacts of traffic and transport on the environment (Van Wee and Handy (2016)). The municipality of Amsterdam strives for cleaner transport alternatives, such as cycling and public transport, reduction of cars on the streets and reduction of parking locations to create more public spaces (Dijksma (2020)). In order to support the decision-making of these policy measures, information about the environmental factors that influence the parking behavior and the predicted parking occupancy is required. The objective of this chapter is to propose a forecasting model that provides insight into the future off-street parking occupancy.

For this study, historic off-street parking occupancy information of 57 public parking garages and P+R locations in Amsterdam is used. These parking locations are either offered and controlled by the municipality of Amsterdam or by commercial companies, and cover over 90% of the total public off-street parking locations in Amsterdam. The parking occupancy data are obtained from open data feed and are retrieved from the barrier systems in the parking location. Using this data, a model is presented that gains insight into the factors affecting the parking choice. We compare 3 different temporal models to predict the parking occupancy for six months ahead: Long Short Term Memory (LSTM), Seasonal Autoregressive Integrated Moving Average with Exogenous variables (SARIMAX) and Exponential Smoothing models (ETS). The benchmark model is a Seasonal Naive model. Additionally, we present a software system to visualize the predicted parking data on a map, allowing decision makers to consult the occupancy rate

forecast for six months ahead.

3.2. BACKGROUND

The prediction of vehicle parking occupancy and urban land use and transport planning have been studied extensively during the past decade.
Manville, Shoup, et al. (2010) studied the interplay of parking with the built environment in the City of Los Angeles, and concluded that eliminating the minimum standard led to a 40–55% reduction in the parking supply. Christiansen et al. (2017) analyzed the impact of parking availability at the start and the end of a trip on mode choice. They concluded that parking restrictions at the workplace are more effective in reducing car-use on work trips than the regulation through parking fees. Liu et al. (2017) investigated the interaction of parking with the built environment in densely built neighborhoods in China. They found that the parking market can actively decrease the residential parking ratio, that minimum parking standards lead to more office parking in dense neighborhoods, and better transit services and lower parking ratios are supported. However, the increase of office parking in these neighborhoods counteracts the benefits of compact development. De Gruyter et al. (2020) explored the association of high quality public transport with reduced car parking requirements in Melbourne. They concluded that a 10% increase in public transport service supply is associated with a 0.9–1.2% reduction in car parking demand.

In the field of parking occupancy prediction, Stolfi et al. (2017) compared various models on parking prediction in the city of Birmingham, and found that the time series method provided the most accurate results. They also presented a web page prototype to visualize the current and historical parking data on a map. Differences with their study is that our study models for a larger time span and incorporates external factors. S. Yang et al. (2019) proposed a real-time model using deep learning approaches such as LSTM for occupancy forecast in the Pittsburgh downtown area. For input, they used heterogeneously structured traffic data sources, such as parking meter transactions, traffic speed, and weather conditions. They concluded that incorporating traffic speed and weather information significantly improves the parking predictions.
In various time series analysis studies on parking space prediction, variants of ARIMA have achieved valuable results. For example, Zhu et al. (2018) applied an ARIMA model with additional real-time short-term forecasting framework to create a parking guidance system in Nanjing China, that outperforms a conventional neural network method and the Markov chain method. Friso et al. (2017) implemented Seasonal ARIMA in a short-term traffic prediction case study that, despite its simplicity, obtained more accurate results than more complicated methods like multivariate spatial-temporal ARIMA. A major difference is that these studies are more supposed for real-time purposes and predict one-step ahead, while the purpose of this research is to gain insight into the parking occupancy in a larger time span.

The following studies highlight the most important factors that influence car drivers'

parking decisions. According to Kaplan et al. (2009) historic parking data is influenced by past parking availability for predetermined time intervals, days of the week, weather conditions and events. Other influential factors found by van der Waerden and Oppewal (1996) are parking cost and walking distance. Likewise, Khaliq et al. (2018) found that the key attributes which the driver considers are "walking distance to destination" and "parking cost". Golias et al. (2002) studied the off-street parking choice sensitivity and concluded that the difficulty in finding a parking space is a more important factor than the increased parking fee.

Although real time one-step-ahead predictions have been studied widely, only few studies are available on many-steps-ahead predictions. These studies predict the occupancy for one Fan et al., 2018 to two weeks Stolfi et al. (2017) ahead. Contrary to previous studies, this study has a time span of six months ahead. Parking occupancy forecasts of this time span are especially suitable for short-term policy decision making, such as the organization of parking supply for events, for instance by dynamic pricing.

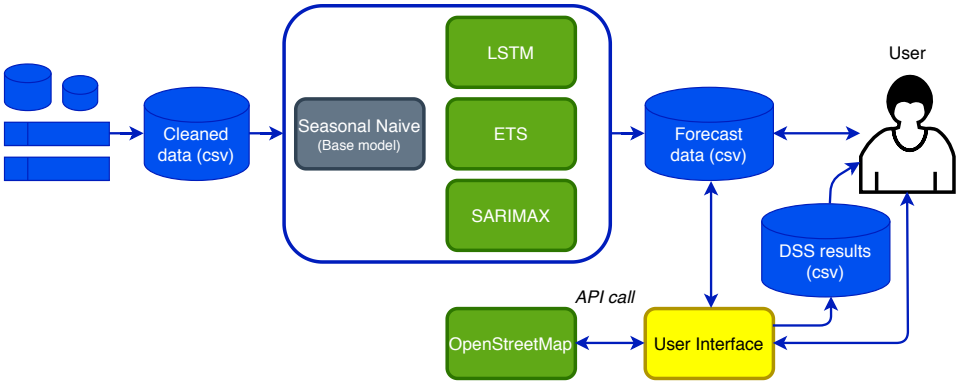


Figure 3.1: System architecture of the forecasting model.

3.3. MODEL SET-UP

The system architecture of the model is given in Figure 3.1. The data set contains information about the occupancy and capacity per minute for the 57 parking locations in Amsterdam during 2018 and 2019. This data set contains warnings, errors and missing values. First, these are detected and imputed using Kalman Filter imputation (Welch, Bishop, et al. (1995)). Afterwards, the parking occupancy is aggregated into hourly data using the robust median. The cleaned data set is used to compare the LSTM, SARIMAX and ETS models, with a Seasonal Naive model as a benchmark model. The hourly forecasts of the best performing model can be loaded into the Decision Support System (DSS). The goal of the DSS is to provide a clear insight into forecast parking occupancy in order to assists a human decision maker to make choices on parking more effectively. The user can work with this system interactively, by giving input on certain parking locations and time ranges and save the results. The user interface uses the graphics from

OpenStreetMap to visualize the off-street parking locations on a map.

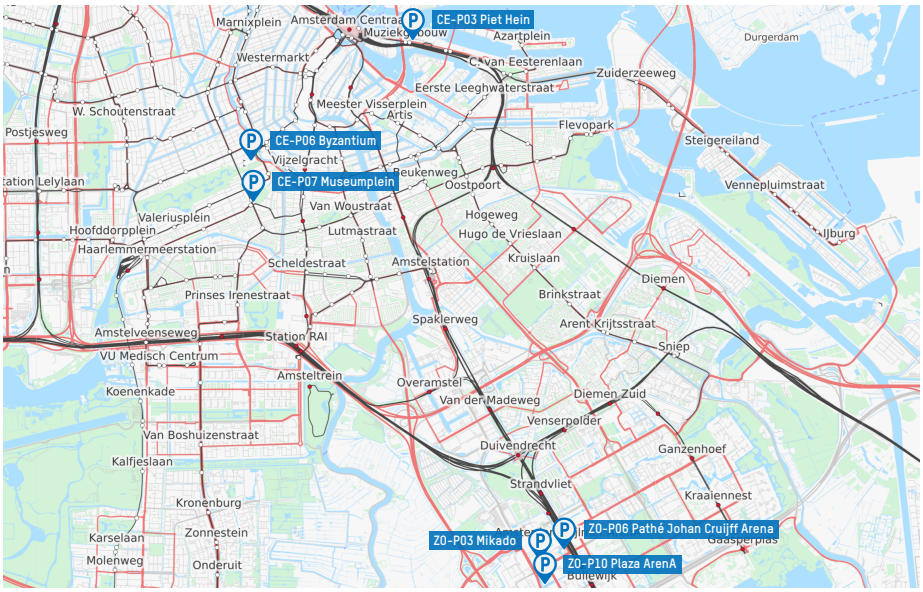


Figure 3.2: Locations of the parking garages in Amsterdam that are highlighted in this chapter.

3.4. CASE STUDY

For the visualization in this chapter six parking locations are highlighted, these are indicated on the map in in Figure 3.2. The garages ZO-P03 Mikado, ZO-P06 Pathé Johan Cruijff Arena, and ZO-P10 Plaza ArenA, are operated in the south east of Amsterdam by the municipality of Amsterdam.

Figure 3.3 visualizes the environment of these parking locations, which contains a soccer stadium, multiple concert halls, and a railway station. Other facilities are an apartment block, a shopping center, and a business park. Therefore, this neighborhood attracts residents and travelers with various purposes. The other three locations, CE-P03 Piet Hein, CE-P06 Byzantium, and CE-P07 Museumplein, are located in the city center of Amsterdam. The environment of the first, CE-P03 Piet Hein, is given in Figure 3.4.

This parking location, operated by the Municipality of Amsterdam, is based near the Amsterdam Central railway station, a concert hall, a ferry terminal, and the historic city center. The last two are stationed in the historic city center and are operated by Q-Park, a commercial parking operator.

Figure 3.5 shows that these parking locations are stationed nearby a park, museums, concert halls and other entertainment venues. In addition, multiple commercial streets are located in this area.

Figure 3.6 visualizes the development in occupancy and capacity of these locations during Monday, July 24 midnight, 2019 to Sunday, July 30 midnight, 2019. Note that the capacity can change over time. This is because some parking locations in Amsterdam

Figure 3.3: Spatial environment of the parking garages in south-east of Amsterdam.

Figure 3.4: Spatial environment of CE-P03 Piet Hein.

Figure 3.5: Spatial environment of CE-P07 Museumplein and CE-P06 Byzantium.

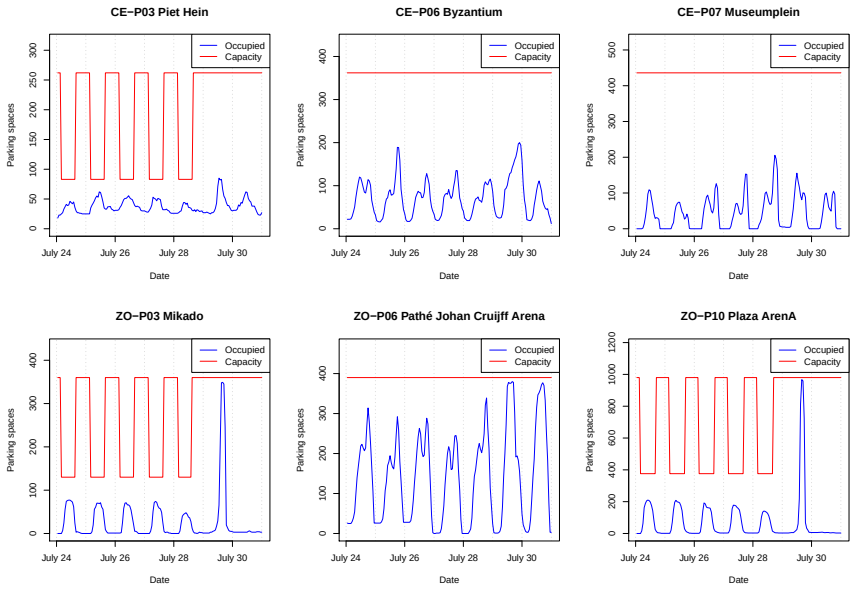


Figure 3.6: Occupancy and capacity of six parking locations, July 24-30, 2019.

work with time-varying parking regulations. It can also be observed that the parking spaces are less occupied to idle overnight, whereas at daytime or in the evening a clear peak is shown. The graphs of CE-P06 Byzantium, CE-P07 Museumplein and ZO-P06 Pathe Johan Cruijff Arena have double peaks. These are typical for parking locations that serve both commercial and recreational purposes. Visitors park for commercial purposes around noon, and park for catering establishments and entertainment venues in the evening. In two parking locations established near the soccer stadium and concert halls (ZO-P03 Mikado and ZO-P10 Plaza ArenA) a larger peak (e.g. Saturday, July 29) can occur, which is caused by a soccer match.

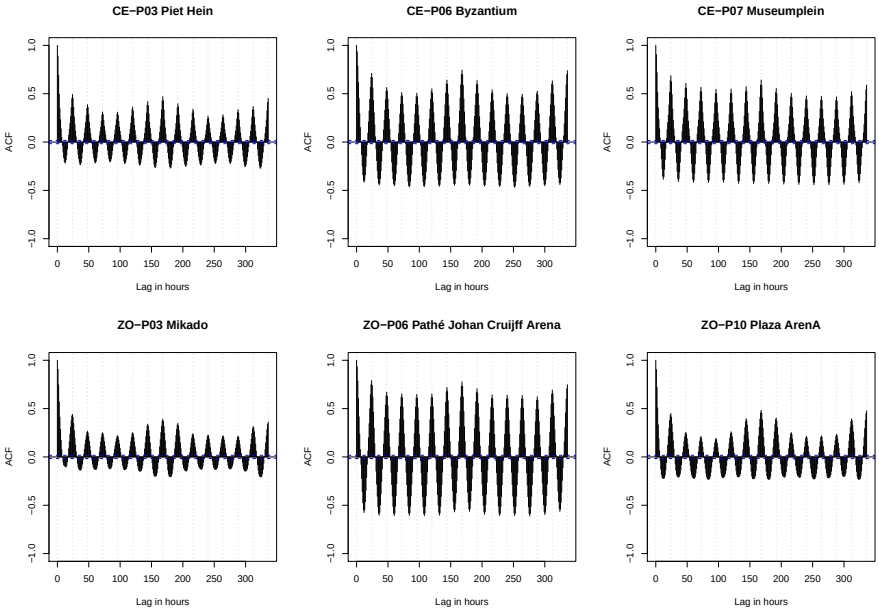


Figure 3.7: Empirical study: Autocorrelation function of six locations.

Patterns in the data set can also be investigated using autocorrelation function (ACF) plots, given in Figure 3.7. The values on the x-axis are the lags λ which measure the correlation between a time series and a shifted variant (by λ time steps) of itself. This provides clues to an underlying model that describes the data well and characterizes the predictability of a time series (Hoogendoorn and Funk (2018)). At the beginning of the ACF plots high correlations are observed. This is because the lower shifted version of itself is highly correlated with the current value (i.e. the lag λ is close to 0). Further, other higher peaks represent daily and weekly components. From these plots, both intraday and intraweek patterns can be observed: each day (24 lags in hours, visualized with vertical grid) has a peak; additionally, each week (168 lags in hours) has a larger peak. Note that some locations have a stronger intraday and intraweek pattern (e.g. CE-P06 Byzantium) than other locations (e.g. ZO-P03 Mikado). Thus, from Figure 3.6 and 3.7 clear patterns, but also differences between the locations can be found. Therefore, the park-

ing occupancy forecasts are modeled for each parking location separately, rather than applying one general model.

3.5. IDENTIFICATION OF FACTORS AFFECTING OFF-STREET PARKING OCCUPANCY

This section highlights the statistical significance of external factors on the parking occupancy. The factors examined are the weather, events, parking tariffs, and the addition of a public transport line. Factors that contribute significantly to the model performance are implemented in the forecast model. In Table 3.1 a summary of the temporal variables is given.

3.5.1. WEATHER

The weather metadata is provided by the Royal Netherlands Meteorological Institute (KNMI) from Schiphol airport station near Amsterdam. This data set contains hourly observations for 24 weather related variables, such as temperature, air pressure and rain. Some of these variables are highly dependent on each other. For instance, the horizontal view has a negative relationship with fog. To avoid overfitting, a selection is made in a way that the weather variables are not explained by any other variable. The remaining weather variables are average wind speed, temperature, sunshine, precipitation, view and thunderstorm.

3.5.2. EVENTS

According to previous studies, sports and artistic events substantially affect the parking occupancy in nearby parking locations (Arthur (1955) and Grodi et al. (2016)). This chapter focuses on the impact of sports and artistic events in the stadium Johan Cruijff ArenA and the concert halls Ziggo Dome and AFAS Live, located in the south-east of Amsterdam. For each event type, nine binary variables are included in the model. For sports events, four binary variables are added for the hours before the match starts ("Pre-match" in Table 3.1). These variables have a 1 if the certain hour is respectively 1, 2, 3, and 4 hours before a match takes place and 0 otherwise. Similarly, four binary variables are added for respectively 1, 2, 3, and 4 hours after the match is over ("Post-match" in Table 3.1). Finally, a binary variable is included which is 1 during the match and 0 otherwise ("During match" in Table 3.1). Nine variables are added instead of one, since in reality the amount of occupied parking spaces does not increase to its maximum in one hour before an event starts and drops back to its original value after the event is over. Instead, visitors arrive and leave a parking location more gradually. This is caused by both the limitations of the parking system, and the crowded traffic on the surrounding roads. For these reasons, a maximum of approximately 250-300 cars can leave in one hour per barrier. With the same approach, music events are added to the model ("Pre-concert", "During concert" and "Post concert" in Table 3.1).

3.5.3. PARKING TARIFFS

In almost all major cities, parking pricing is an important consideration for policy makers (Kelly and Clinch (2006)), because it does not only significantly contribute to the

Table 3.1: External attributes for effect analysis.

Attribute name	Explanation
Weather attributes	
Average wind speed	Average wind speed (in 0.1 m/s) for the previous 10 minutes
Temperature	Temperature (in 0.1 °C) on 1.50m height during observation
Sunshine	Duration of the sunshine (in 0.1 hours) per hour calculated from global radiation (-1 for <0.05 hours)
Precipitation	Hourly sum of the precipitation (in 0.1 mm) (-1 for <0.05 mm)
View	Horizontal view during observation. 0 = less than 100m, 1 = 100-200m, 2 = 200-300m, ... , 50 = 5-6km, 56 = 6-7km, 57 = 7- 8km, ... ,79 = 29-30km, 80 = 30-35km, 81 = 35-40km, ..., 89 = more than 70km
Thunderstorm	0 = did not occur; 1 = did occur in the previous hour
Event attributes	
Pre-match	For h = 1, 2, 3, 4 hours before a sports match 0 = not h hours before a sports match; 1 = h hours before a sports match
During match	0 = not during a sports match; 1 = during a sports match
Post-match	For h = 1, 2, 3, 4 hours after a sports match: 0 = not h hours after a sports match; 1 = h hours after sports match
Pre-concert	For h = 1, 2, 3, 4 hours before a music event 0 = not h hours before a music event; 1 = h hours before a music event
During concert	0 = not during a music event; 1 = during a music event
Post-concert	For h = 1, 2, 3, 4 hours after a music event 0 = not h hours after a music event; 1 = h hours after a music event
Parking tariff attributes	
Parking tariff hour	The parking tariff at the relevant parking location, at the given hour
Maximum day tariff	The maximum parking tariff for a day ticket at the relevant parking location
Public transport line attribute	
North/South metro line	0 = before the North/South metro line opening; 1 = after the North/South metro line opening

parking occupancy (Khaliq et al. (2018)), it influences the performance of the whole transportation network (Jakob et al. (2020)). However, according to the municipality of Amsterdam, even though parking in garages is 15% cheaper than on-street parking in the same neighborhood, no strong (relative and absolute) increase in off-street parking occupancy is detected yet. Paid parking always applies in the locations investigated in this study. The local variation over time in parking prices is minimal. Two of the examined locations have a time-varying parking tariff, but the other four parking locations have a constant parking tariff which is insensitive to the hour of the day. One change in these variables occurred on July 14, 2019. From this day, the parking tariffs have increased drastically, sometimes with 100% (Kruyswijk (2018)). For each parking location, the parking tariffs in the certain hour and the maximal parking price for one day are added to the model.

3.5.4. PUBLIC TRANSPORT LINE

As an experiment, a new metro line is included as a variable. This metro line is the North/Southline, which connects the north to the south of Amsterdam via the city center. A 0 is given for the period before the start of the adjusted timetable by the new metro line on 22 July, and a 1 from the start of the adjusted timetable, from 22 July.

3.5.5. RESULTS OF THE EFFECT ANALYSIS

For the effect analysis, the estimates and p-values from a z-test of a SARIMAX model are used. This model contains autoregression and moving average processes and the exogenous regressors in Table 3.1. Table 3.2 shows the estimates and the p-values of the weather, event, and public transport line attributes that significantly affect the parking occupancy. Statistically significant variables not included in the table are the autoregressive and moving average component variables. The parking tariff variables are not included in the table, because these appear to be not statistically significant for the parking occupancy prediction of the examined parking locations. This contradicts with the literature such as Jakob et al. (2020), Kelly and Clinch (2006), and Khaliq et al. (2018). An explanation for this is that the local variation over time in the parking tariff prices is minimal. In four of the parking locations the parking tariffs only change once from April 14, 2019. This parking tariff change is at the end of the time range of the data set. Thus, the model does not have enough data to learn from the changed parking tariffs.

Focusing on the weather attributes, it is found that the average wind speed has a modest negative impact on the occupancy in the locations ZO-P06 Pathé Johan Cruijff Arena and CE-P06 Byzantium (estimates of -0.066 and -0.041, respectively). The less wind, the slightly more people park their car in the above-mentioned locations. In half of these parking areas, temperature has a small negative correlation with the parking occupancy. An explanation for this is that on a colder day travelers choose the car over an alternative transport mode (e.g. bicycle or public transport). Significant weather variables in CE-P06 Byzantium are precipitation (which has a negative impact) and sunshine (which has a positive impact). This makes sense as this garage is located close to a park and commercial streets. A sunnier and less rainy day can attract more visitors to the city center. One variable that has a strong negative impact on parking is thunderstorms (-9.194 for ZO-P06 Pathé Johan Cruijff Arena). A speculation is that during a thunderstorm,

potential travelers choose to wait for the storm to pass. Except for thunderstorms, the absolute difference in predicted occupied parking spaces caused by the weather is quite small. However, these results provide evidence that seemingly unimportant factors can still slightly affect the parking occupancy. For policy decision makers this means that including these external factors can help in predicting the parking occupancy more accurately. The weather variables are not relevant for six months ahead policy making, but can be insightful for policy making for a few hours ahead (e.g. during events and holidays).

The event attributes strongly affect the parking occupancy, especially in the parking locations nearby this event location (ZO-P03 Mikado, ZO-P10 Plaza Arena and ZO-P23 Bijlmerdreef). Also in the city center (CE-P07 Museumplein) a significant effect is noted during the match until one hour after the match. This impact can be caused by this match, but also by events that could occur at the same time in a concert hall or music hall close to this parking location. To find the exact source of this increase, variables with events in the city center should also be added to the model. The increase caused by a sports match often starts 3 hours before the match, reaches its peak during the match, and stays high for 2 up until 4 hours after the match. When the music concert estimates are compared to the match estimates, we can see that in absolute numbers less people park for a music event compared to a match. Possible reasons can be that music event visitors choose a different mode (e.g. public transport), or that the concert variable does not distinguish big concerts (in the football stadium) from smaller concerts (in a smaller music venue). It can also be noted that after a music event visitors directly go home, compared to a match event, where visitors stay up until 4 hours after the match has ended.

Finally, the North/South line has a negative impact on parking in ZO-P10 Plaza ArenA. Since the metro line started operating, the number of parked vehicles in this location has dropped significantly. A possible explanation is that the metro line improved connectivity with the city center, but is not connected near this parking location. Accordingly, this line may motivate the car to transit travelers who originally parked in ZO-P10 Plaza ArenA to park elsewhere. It is however unlikely that this affects a large absolute number of parked vehicles, because the number of short term parkers is relatively small compared to parking subscription holders in Plaza ArenA. In order to find explanations, more research needs to be conducted into this phenomenon.

Although some attributes affect the parking occupancy in many parking locations, such as the match variables, clear differences between the locations can be noted. For instance, the location CE-P03 Piet Hein is not affected by any of the external variables. A reason for this is that mainly subscription and permit holders park at this location. In addition, there are hardly any events close to this location that attract a large audience. Further, because the parking location is close to the central station, destinations in the same neighborhood can also easily be accessed by bicycle and public transport. Because the exogenous regressors are location-specific, the significant factors on parking are investigated per location.

Table 3.2: Significant external factors

Attribute	**ZO-P03 Mikado**		**ZO-P06 Pathé Johan Cruijff Arena**		**ZO-P10 Plaza ArenA**	
Weather attributes	*Estimate**	*p-value***	*Estimate*	*p-value*	*Estimate*	*p-value*
Avg. wind speed	-	-	-0.066	0.040	-	-
Temperature	-0.072	4.504e-5	-0.064	0.040	-	-
Precipitation	-	-	-0.156	0.023	-	-
Thunderstorm	-	-	-9.194	0.008	-	-
Event attributes	*Estimate*	*p-value*	*Estimate*	*p-value*	*Estimate*	*p-value*
Pre-match (1h)	145.611	<2.2e-16	30.163	1.850e-6	360.638	<2.2e-16
Pre-match (2h)	81.251	<2.2e-16	27.024	7.318e-5	189.587	<2.2e-16
Pre-match (3h)	27.552	<2.2e-16	21.022	3.377e-5	44.888	8.906e-11
During match	173.548	<2.2e-16	37.119	6.611e-12	476.499	<2.2e-16
Post-match (1h)	82.557	<2.2e-16	-	-	262.391	<2.2e-16
Post-match (2h)	16.340	0.0004	-25.752	1.821e-12	94.311	<2.2e-16
Post-match (3h)	6.768	0.045	-	-	51.418	1.277e-7
Post-match (4h)	-	-	-	-	17.199	0.013
Pre-concert (1h)	16.585	6.754e-11	19.803	2.311e-6	36.119	6.596e-15
Pre-concert (2h)	10.475	8.198e-5	25.009	2.759e-7	14.486	0.0002
Pre-concert (3h)	7.558	0.0003	23.901	2.651e-7	-	-
Pre-concert (4h)	-	-	14.211	1.176e-5	-	-
During concert	16.463	2.543e-15	19.060	5.805e-9	35.112	<2.2e-16
Public transport line attribute	*Estimate*	*p-value*	*Estimate*	*p-value*	*Estimate*	*p-value*
North/South line	-	-	-	-	-12.827	0.042
Attribute	**CE-P03 Piet Hein**		**CE-P07 Museumplein**		**CE-P06 Byzantium**	
Weather attributes	*Estimate*	*p-value*	*Estimate*	*p-value*	*Estimate*	*p-value*
Avg. wind speed	-	-	-	-	-0.041	0.007
Sunshine	-	-	-	-	0.143	0.029
Event attributes	*Estimate*	*p-value*	*Estimate*	*p-value*	*Estimate*	*p-value*
During match	-	-	19.761	1.625e-08	-	-
Post-match (1h)	-	-	20.019	8.104e-12	-	-

* The estimates represent the correlation with the parking occupancy, where a value < 0 signifies a negative correlation and a value > 0 signifies a positive correlation.
** The p-values are obtained from a z-test. Using significance level $\alpha = 0.05$ the following hypotheses are considered: $H_0 : \beta_i = 0$ versus $H_1 : \beta_i \neq 0$ for $i = 1, ..., 28$. The null-hypothesis is rejected for the attributes with a p-value < α. The coefficients with p-value > α are not statistically significant (indicated with "-").

3.6. TEMPORAL FORECASTING MODELS

In this section five forecasting approaches are described: Seasonal Naive, Seasonal autoregressive integrated moving average (with and without exogenous variables) and Error, trend, seasonality models and a Long short term memory neural network. To compare the predictive performance of these models and to reduce overfitting, the measurements of the parking locations are split into a training, validation and test set in temporal order. These are 60%, 20% and 20%, respectively.

3.6.1. KEY PERFORMANCE INDICATOR

For model comparison, the Key Performance Indicator (KPI) Root Mean Squared Error (RMSE) is chosen based on two reasons. First, the RMSE does not divide by the target variable (i.e. the amount of occupied parking spaces) which is often 0. This would make for instance the Weighted Mean Absolute Percentage Error a less suitable KPI for the present case study. Second, the KPI is more strict when extreme peaks in the actual values are missed by the models. This is essential as these moments can indicate a fully occupied parking location. The equation is as follows

$$RMSE = \sqrt{\frac{\sum_i (\hat{y}_{i+\tau} - y_{i+\tau})^2}{N}} \tag{3.1}$$

with $y_{i+\tau}$ the actual occupancy, $\hat{y}_{i+\tau}$ the predicted occupancy for $i = 1, ..., N$ samples, with τ the time step.

3.6.2. SEASONAL NAIVE

Because the parking occupancy data shows a strong seasonal pattern, the Seasonal Naive Model is chosen as a benchmark approach. This model is simple and requires minimal data, but tends to be quite inaccurate at predicting highly fluctuating data or data that changes due to irregular factors. Because of this, the expectation is that this model will not work extremely well. In Seasonal Naive, a forecast value equals the last observed value of the same season of this data point. The h-step ahead forecast is given in the following equation

$$\hat{Y}_{t+h|t} = Y_{t+h-s(k+1)} \tag{3.2}$$

where h denotes the step size, s is the seasonal period (Barak et al. (2019)) and k stands for the number of complete time steps in the forecast period prior to time $t+h$. Because we observed a weekly seasonality for each location, a season of one week is considered in this model.

3.6.3. SEASONAL ARIMA WITH EXOGENOUS REGRESSORS

An ARIMA(p,d,q) model is based on a combination of autoregression AR and moving average MA processes of order p and q respectively (Badii et al. (2018)). To make time series with a trend component stationary, the model also differences over the trend (order d). An extension is seasonal ARIMA(p,d,q) (P,D,Q), which differences over the seasons of the corresponding parameters p, d and q as well. The addition of seasonal components to the model is highly suitable for the seasonal parking data. This model is extended further by implementing the temporal exogenous variables discussed in the previous section, resulting in a SARIMAX model. SARIMAX models are a combination of regression and SARIMA. In the last few years, these models have been widely used in previous traffic demand forecast studies, for example in Cools et al. (2009) and Fışkın and Cerit (2019) and Lin et al. (2013). The step-up method is applied to build the model. In a step-up method, we start with an empty model, and step-wise add one statistically significant exogenous regressor. For hyper-parameter testing, grid search is applied on each separate parking location. Because the parking locations show a different time de-

velopment (see section: Empirical Study) and statistically significant exogenous regressors (see section: 3.5), the hyper-parameter values p, d, q, P, D and Q and the regressors vary as well based on the location.

3.6.4. ETS MODELS

Error Trend Seasonality (ETS) models are classic time series models. These models predict based on applying exponential smoothing on the three main occurring components in the time series, namely error, trend and seasonality. The chosen model depends on the shape of these error, trend and seasonality components. The error ϵ and seasonality s can be either additive (A), multiplicative (M) or non-existent (N). The trend component b can either be additive (A), additive damped (Ad) or non-existent (N). Based on the characteristics of the respective parking location time series, the suitable equation can be chosen in Table 2.1 in the book by Hyndman et al. R. Hyndman et al. (2008). The parameters γ and β are the smoothing parameters of the seasonal and trend component respectively, and Φ is the damp parameter that reduces the trend each time period for the damped additive model. Similar to ARIMA, grid search per parking location is applied, in order to find the correct specific ETS model.

3.6.5. LONG SHORT-TERM MEMORY

Long Short-Term Memory neural network (LSTM) is a variant of a recurrent neural network (RNN). RNN can be used to approximate almost any dynamic system. Traditional RNN are trained via a gradient based approach. This approach has the fundamental problem that the back propagation steps are exponentially dependent on the depth of the weights (Gers et al. (1999)), resulting in vanishing or exploding gradients. Therefore, traditional RNN are unable to detect dependencies for 10 or more steps. LSTM enforces the backpropagation errors to be constant. Hence the gradients cannot explode or vanish. Figure 3.8 shows one hidden layer of an LSTM model. This layer contains one memory cell and two adaptive multiplicative gating units in which each has an input and an output gate. These gates give and receive information to all memory cells in the block. The memory cells contain a recurrently self-connected linear unit called the "Constant Error Carousel" (CEC). This solves the vanishing gradient problem, since when there is no input at a certain point in the data or when there are errors in a cell the backpropagation steps remain the same. When the activation of a CEC gets close to 0 the irrelevant data and noise do not disturb the other memory cells. The CEC activation function takes three input variables and its own prior state, namely: net_c (the ingoing input in the cell itself), net_{in}, net_{out}, which are the respective inputs and outputs of the output gates of the gating units. For this model it is assumed that the predictions will be a discrete timestamp ahead. For each step the weights of all units and the CEC need to be updated. The activation function used for the input and output gates is the standard sigmoid function. The input that the memory cell receives from itself is then reduced by a centered sigmoid function. An LSTM model does not assume the amount of data and has the benefit that it does not necessarily need to have a trend or seasonal component in order to predict something without violating the assumption.

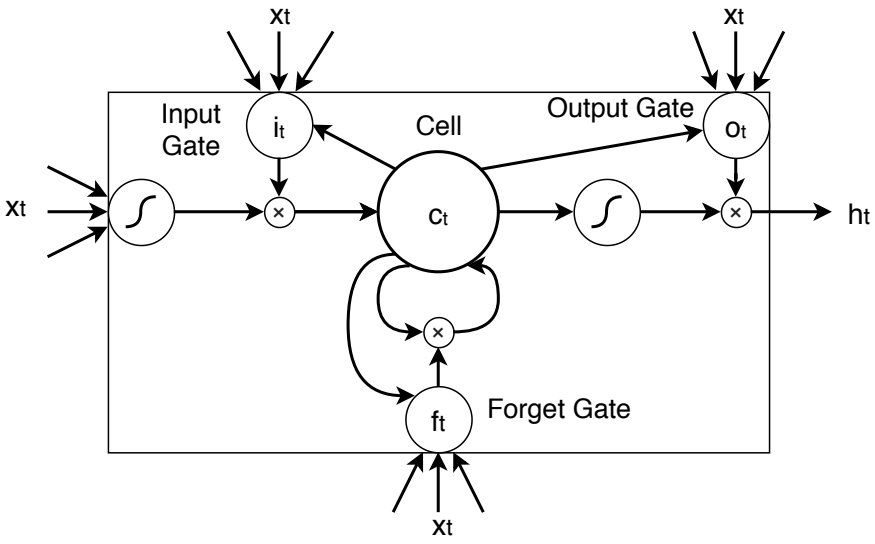


Figure 3.8: One Hidden Layer of an LSTM Model [adapted from Gers et al. (1999).]

Table 3.3: RMSE on Seasonal Naive, ETS, SARIMA and SARIMAX, six months ahead forecast and LSTM, 1 hour ahead forecast.

Timespan	Model	ZO-P03	ZO-P06	ZO-P10	CE-P03	CE-P06	CE-P07
6 months ahead	Seasonal Naive	59.481	68.947	96.868	27.970	46.728	71.411
	ETS	56.295	65.331	92.037	**26.234**	44.221	55.308
	SARIMA	59.919	68.932	96.868	28.030	47.743	71.195
	SARIMAX	**51.208**	**60.258**	**70.767**	28.030	**43.733**	**51.891**
1 hour ahead	LSTM	0.656	2.179	1.803	0.202	1.104	1.061

3.7. RESULTS OF THE FORECASTING MODELS

In Table 3.3 the models are compared based on the lowest RMSE value in the validation set. LSTM is not included in the six months ahead forecasts due to the extensive input data set and limited memory space. Instead, an alternative model with one hour ahead forecasts is proposed. Note from the table that the LSTM model obtained lower RMSE values. The improved results can be explained by the fact that for one hour ahead we have more knowledge about the recent actual values, whereas for six months ahead the predictive power reduces over time. Based on the lowest RMSE of the six months ahead predictions, the SARIMAX model outperformed the other models, except for CE-P03 Piet Hein, where the ETS model is the most accurate. Because none of the external regressors contributes significantly for this location, the SARIMAX and SARIMA RMSE values are identical. The highest errors were obtained by the benchmark model Seasonal Naive (in the South-East of Amsterdam, ZO-P03, ZO-P06, ZO-P10), and by the SARIMA model

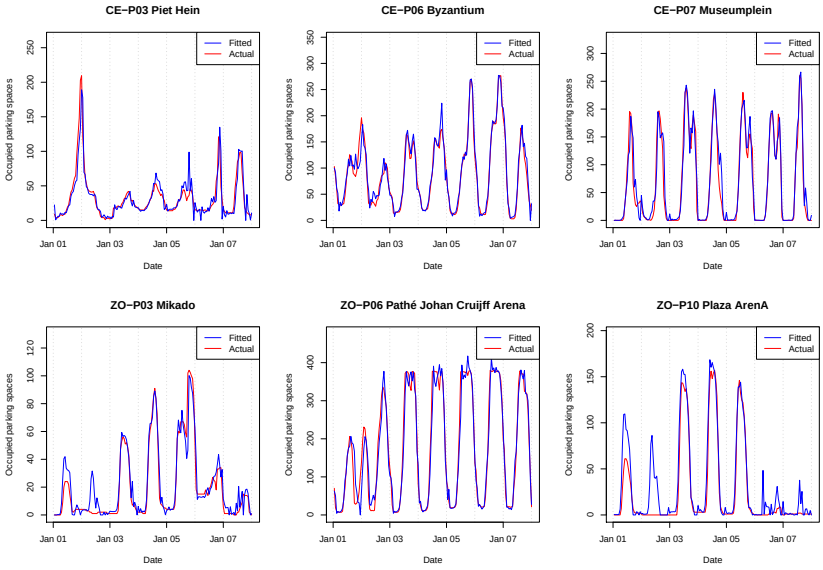


Figure 3.9: SARIMAX model, fitted values.

(in the city center). A possible explanation for the less accurate results by the SARIMAX model in the center is that parking in the city center is less predictable compared to the south-east of Amsterdam. When SARIMA and SARIMAX are compared, strong differences in RMSE can be noted. For example, adding the external regressors in Plaza ArenA and Museumplein decreased the RMSE values by 27%. For CE-P06 Byzantium, the addition of the speed and sunshine variables reduced the RMSE from 47.743 to 43.733, which is an improvement of about 8%. Although this difference is smaller than in Plaza ArenA and Museumplein, inclusion of weather information improves the prediction performance.

Figure 3.9 presents the fitted values of the SARIMAX model for the six locations in between January 1, 2020 to January 7, 2020. From this figure we observe that the model and the external factors are close to the actual values.

Figure 3.10 compares the six months ahead forecasting models, for the location Plaza ArenA during January 19, 2020 to January 25, 2020. For many steps ahead forecasts, each of the models predicts the seasonal patterns well. The SARIMAX model is the only model that is able to predict the off-seasonal spikes on January 20 and January 24. These spikes are events, thus are not a part of a daily or weekly pattern. Because exogenous regressors can be included in a SARIMAX model, these events can be incorporated. Note that the first spike is estimated too low and the second spike is estimated too high. If this model has more information regarding the expected number of visitors, these results can be improved even more.

Figure 3.11 visualizes the six months ahead SARIMAX predictions for six months ahead predictions for the six locations. Except for CE-P03 Piet Hein, where the exter-

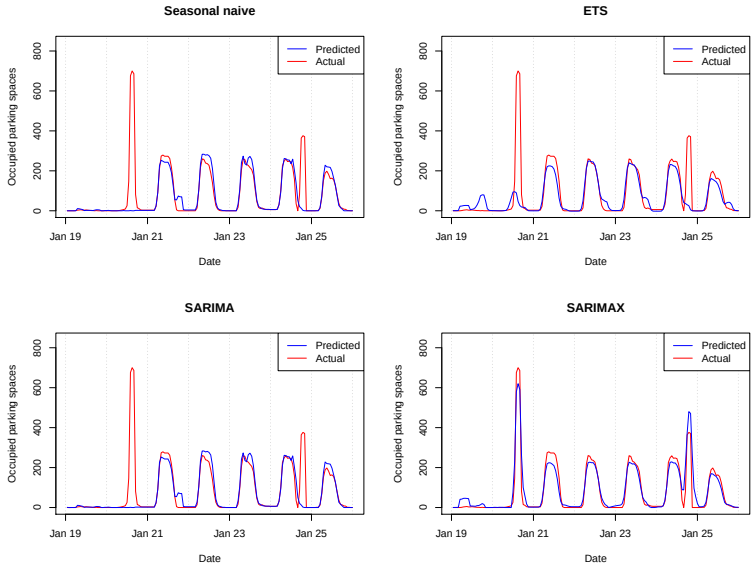


Figure 3.10: Comparison of models with six months ahead forecasts.

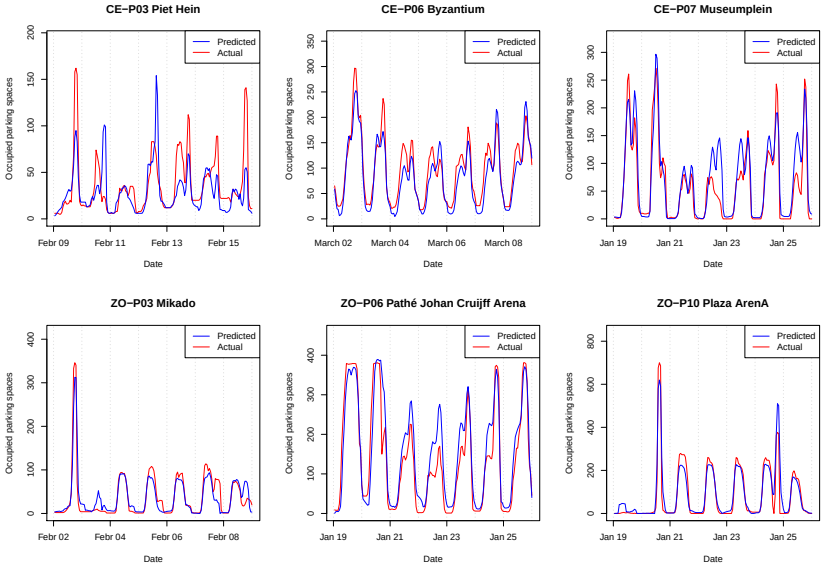


Figure 3.11: Comparison of actual against forecasted values for the SARIMAX predictions.

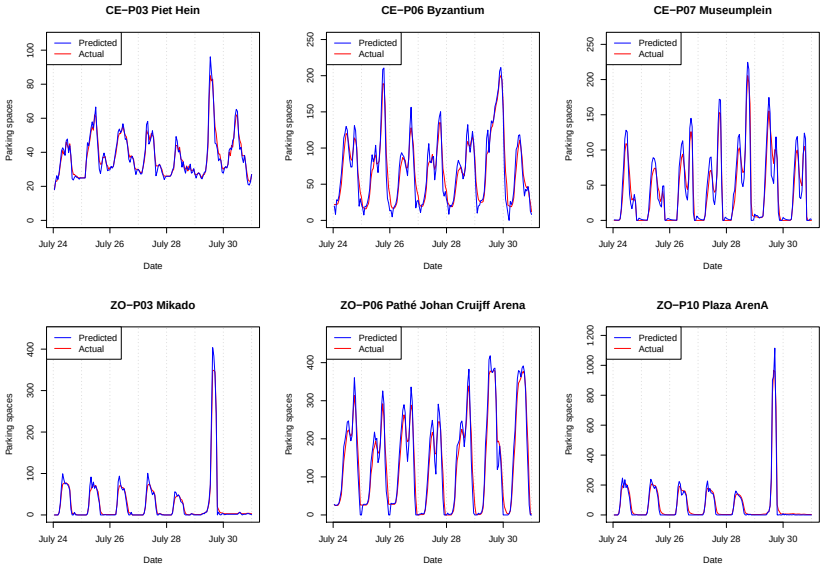


Figure 3.12: Comparison of actual against forecasted values for the LSTM predictions, one step ahead.

nal regressors did not have any impact, the SARIMAX model is able to predict the daily patterns fairly well, especially for the broad timespan.

In Figure 3.12 a similar graph is given for the LSTM model results with one hour ahead predictions. This model seems to detect the complex structures and patterns in the data more than the other models. With smaller step sizes, the model outperforms the multi-step ahead predictions of the other algorithms.

GRAPHICAL USER INTERFACE DECISION SUPPORT SYSTEM

The Decision Support System (DSS) provides information on the parking occupancy forecast. For most of the locations, the SARIMAX model, and for a few locations (e.g. CE-P03 Piet Hein) the ETS model is implemented in the DSS. The user interface of the DSS is presented in Figure 3.13. Above the dotted line the main user interface of the DSS is visualized. On the left side of the user interface the decision maker can select the parking locations, and the date and time range of interest. On the right side of the user interface a map is provided. This map uses the graphics of OpenStreetMap, where the values in the placemarks represent the predicted parking occupancy rate for each parking location, at the selected time. The results screen is given below the dotted line. The time series graphs of the selected locations are visualized, and the moments of expected occupancy greater than 95% and statistics are given. On top of the screen, the decision maker can save these results with “Export to CSV” and select new data with “Select Data”. The DSS can support the policy maker or parking manager to ultimately reduce the urban congestion problems mentioned in the introduction. Instances of applications are:

1. *Dynamic pricing*: One might increase the parking prices per hour during times

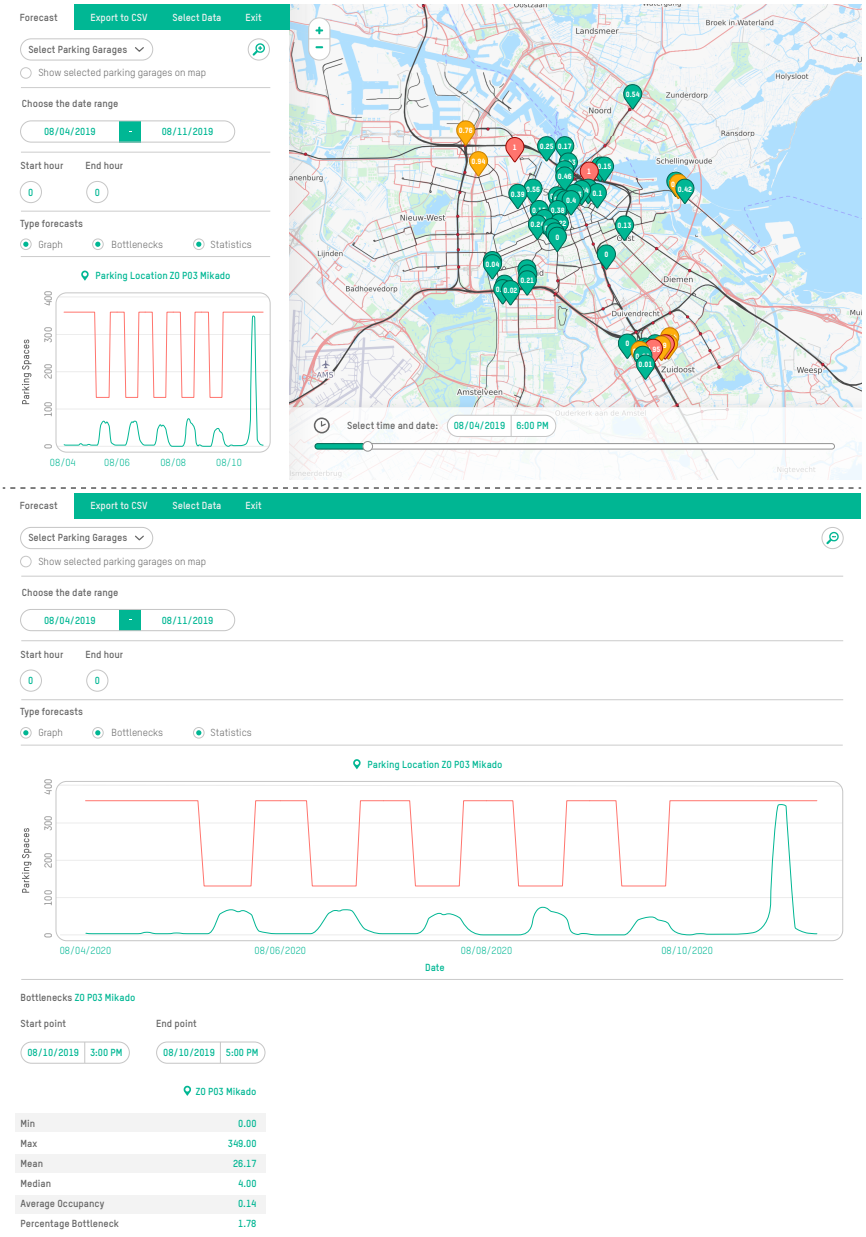


Figure 3.13: User interface of the Decision Support System with Parking Occupancy Forecasts.

when the parking occupancy is expected to be high. For instance, if the parking occupancy is expected to be over 85%, the prices per hour can be raised to €1.00, and if the expected parking occupancy is under 40%, the prices per hour can be lowered by €1.00. The goal of this strategy is to reduce the auto mode share during peak hours, and subsequently prevent congestion problems such as "cruising for parking" (here, an occupancy of 15% is found to be optimal to minimize cruising for parking Arnott and Inci (2006)). One can also use the DSS forecasts as an input for other existing dynamic pricing models Lei and Ouyang (2017) and Tian et al. (2018).

2. *Urban planning*: If it is noted that a parking lot is expected to be (almost) vacant for a long period of time, these can be replaced with public areas (e.g. city parks) and cleaner traffic infrastructure (e.g. bicycle lanes).

3.8. CONCLUSION

The objective of this chapter is to propose a forecasting model that provides insight into the future off-street parking occupancy. To meet this objective, five forecasting algorithms are compared, namely Seasonal Naive Model, Box-Jenkins Seasonal Autoregressive Integrated Moving Average, with and without exogenous regressors, exponential smoothing and Short-Term Memory (LSTM).

For the six months ahead predictions, the SARIMAX model outperformed the other models for the vast majority of the parking locations based on the Root Mean Squared Error (RMSE). When the RMSE values of SARIMAX are compared to SARIMA, it can be observed that the addition of exogenous regressors reduces the error significantly, sometimes with 27%. This indicates that including external variables is essential for parking occupancy predictions. The external variables which affect the parking occupancy most are the event variables for sport matches and music concerts. Policy makers can use this information to reduce traffic congestion, for example by increasing the parking tariffs when the expected occupancy is highest.

Other external variables that impact the parking occupancy are thunderstorms, average wind speed, temperature, precipitation, sunshine, and the addition of a public transport line. By adding weather variables, the model performance can be improved by 8%, which can be used for a few hour ahead forecasts during events. The parking tariffs did not significantly contribute to the model performance, which may be due to the lack of data and time variability in the parking tariffs. The highest errors are obtained by the benchmark model Seasonal Naive in the south east of Amsterdam, and by the SARIMA model in the center of Amsterdam. For most parking locations ETS models are the second best predictors.

The lowest errors were obtained by the real-time short-term LSTM model, because a shorter prediction step leads to more precise predictions. As a result of this, the model can be extended into an online short-term model, which is especially useful for short-term event policy making. Because LSTM also has the possibility to add external variables, the model performance can be improved further with these extensions. Another improvement on LSTM is to extend the model into a one day ahead predictor, instead of one hour ahead.

Since the event variable has the strongest impact on neighboring parking locations, this variable can be elaborated further in future studies. By including all events around each parking location, more peaks can be detected. Further, information about the number of attendees in an event can be included to improve the model performance.
A follow-up study can be conducted into the parking costs, which in the present study do not seem to have an effect on the parking occupancy. This could be caused by the lack of data after the parking tariff change. To provide stronger conclusions on this topic, more data is needed.

From the data analysis we observed major differences in the parking occupancy development per parking location. To also investigate the spatial effects on the parking choice behavior in future work, spatial features could be added to the models. Examples of spatial features are the walking distance to public transport stations, recreational facilities, business parks, hospitals, and features that define social safety. Similarly, the addition of parking supply design features can provide insightful information for the built environment. An example is the average distance from a parking space to the entrance of the parking lot. If this affects the parking occupancy, this can be incorporated in the design to reduce urban congestion problems. Another instance of a design feature is whether the parking lot is covered and its relation with the weather variables. In order to study the impact of the parking occupancy of neighboring parking locations on a given parking location, also the spatial auto-correlation between neighboring parking locations can be covered in the future.
In this study the effect of temporal features on the parking occupancy is analyzed. Spatial-temporal features would give the opportunity to investigate the effect of changes in traffic and transport on the parking occupancy for different parking locations. Examples of changes that affect the parking occupancy are new public transit networks or the impact of the current policy measures of COVID-19 on parking. This way, we can use information and communication technologies in smart cities to form a strong foundation for urban planners and municipalities, resulting in more accurate decision-making.

II

Part II: Trips and Travel Distance

4

FORECASTING PUBLIC TRANSIT RIDERSHIP USING NEURAL NETWORKS

This chapter provides a framework for the efficient representation and analysis of both spatial and temporal dimensions of panel data. This is achieved by representing the data as spatio-temporal image-matrix, and applied to a case study on forecasting public transport ridership. The relative performance of a subset of machine learning techniques is examined, focusing on Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) neural networks. Furthermore Sequential CNN-LSTM, Parallel CNN-LSTM, Augmented Sequential CNN-LSTM are explored. All models are benchmarked against a Fixed Effects Ordinary Least Squares regression. Historical ridership data has been provided in the framework of a project focusing on the impact that the opening of a new metro line had on ridership. Results show that the forecasts produced by the Sequential CNN-LSTM model performed best and suggest that the proposed framework could be utilized in applications requiring accurate modeling of demand for public transport. The described augmentation process of Sequential CNN-LSTM could be used to introduce exogenous variables into the model, potentially making the model more explainable and robust in real-life settings.

Khalil, S., Amrit, C., Koch, T., & Dugundji, E. (2021). Forecasting public transport ridership: management of information systems using CNN and LSTM architectures. Procedia Computer Science, 184, 283-290.

4.1. INTRODUCTION

In the context of public transport, forecasting ridership is essential for business operations. Knowing the demand for travel allows public transport companies to make the transportation system more efficient, and helps to proactively improve the level of service for their customers and eliminate unnecessary costs (Caceres et al. (2017) and de Oña and de Oña (2015)). Having an effective model of how demand for public transport changes over time is important for future planning, introducing new routes, more efficient schedules, and optimizing transportation operations (e.g. frequency of buses on a certain line) (Jin et al. (2016) and Knoppers and Muller (1995)).

This research aims to improve demand modeling for public transport based on historical ridership data. The main objective of the research is to showcase different combinations of machine learning models in order to effectively account for both spatial and temporal relationships in historical ridership data. These models are trained and then tested on how accurate the forecasts are.

The proposed methodology is a combination of two prominent neural network techniques: Convolutional Neural Networks (CNN) are used in image processing (recognition and classification), and Long Short Term Memory (LSTM) models are great at identifying temporal patterns in data. More specifically, this research provides a performance comparison of 5 different neural network architectures: CNN only, LSTM only, Sequential CNN-LSTM, Parallel CNN-LSTM as well as Augmented Sequential CNN-LSTM architectures. Additionally, a classical statistical baseline approach is showcased in the form of a Fixed Effects Ordinary Least Squares (OLS) model for panel data.

It is not immediately apparent that there would be a substantial difference in performance depending which way the models fit together. Yet, this research will demonstrate that there indeed is significant difference and will provide possible explanations of why this may be the case.

SCOPE OF THE RESEARCH AND RESEARCH QUESTION

The research is conducted on a large spatio-temporal data set counting quarterly ridership across all public transportation operators in the greater Amsterdam region. The scope of the research is limited to a relatively small subset of available Origin-Destination (OD) pairs in the original data set. The main reason for it is the way that the data had been aggregated by the Vervoerregio [1] due to privacy concerns. Only the OD-pairs that have relatively large ridership values (more than 300 rides per day of the week in each quarter in the whole period) have been considered.

Furthermore, this research has been conducted within the framework of the *Impact Study North-South Line*, which means that one of the goals has been to specifically look into how ridership had been affected by the inauguration of the North-South metro Line (NSL) in Amsterdam. The implication of this is that research focuses on routes to and from the North of the IJ.

Also, this research will not focus on exploring *all* possible architectural choices for the CNN, LSTM and combined CNN-LSTM models. Instead, a baseline architecture for all is chosen so that the relative performance of the models in question is highlighted.

[1] Transportation authority of the greater Amsterdam region

It is important to mention that this research does not assume these architectures to be the best machine learning algorithms, but rather such selection of algorithms and combined architectures is based on the review of relevant literature, builds and improves on previous research applying CNN, LSTM and combined architectures to forecasting ridership.

The proposed methodology mainly focuses on efficient input processing. However the described models could easily be adapted to a different forecasting goal, such as, predicting categorical ranges, binary flags, identifying specific cases in data. This would be just a matter of assigning proper labels to a given sample set.

The main research question is to determine whether a CNN could be combined with a LSTM model to effectively extract both spatial as well as temporal relationships from the available ridership data, and if that is the case, showcase which architecture captures these relationships best.

4.2. BACKGROUND

We will describe relevant literature on Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) and combined Neural Networks.

4.2.1. CONVOLUTIONAL NEURAL NETWORKS (CNN)

Convolutional Neural Networks have shown outstanding results in such tasks as image recognition, image classification, and computer vision (Krizhevsky et al. (2017) and Oquab et al. (2014)). One of the advantages of CNN architecture is that convolution layers are sparsely connected to the input, instead of fully connected as in the case of simple neural networks. This allows the possibility for individual filters to extract specific features more efficiently.

Another significant feature of CNN is the introduction of pooling layers, which in the case of Max-Pooling selects the most prominent features in a given sample. These two characteristics (sparse connection to input and pooling layers) drastically reduce the number of modeled parameters and thus allows CNN to be employed on classification problems at a large scale (Karpathy et al. (2014)). At the same time, research in the field of face recognition shows that these characteristics allow CNN to efficiently incorporate spatial relationships in the input data Lawrence et al. (1997).

The CNN model can be described as an automatic feature extractor for different kinds of input data (LeCun and Bengio (1998)). This flexibility of CNN to extract features from different sources is also shown in the field of music information retrieval by T. Li et al. (2010), where CNNs are employed for capturing variations in musical patterns for music genre classification. In fact, authors suggest that there was almost no modification needed to adapt CNN's to this particular task.

In order to apply CNN's for feature extraction for forecasting ridership, the model required some adjustment. Namely, the model inputs would be slightly different since values in the "image" will be representing ridership values at a particular time on a particular route, instead of the level of color intensity for a given pixel of an image. Ridership values needed to be scaled to an interval from 0 to 1 to prevent difficulties in model training.

In the context of forecasting ridership, CNN outputs also need to be conceptualized differently. CNN will be predicting ridership instead of an image class (as is usually the case with image classification). The loss function also had to be adjusted in accordance with the goal of predicting continuous ridership values.

4.2.2. LONG SHORT-TERM MEMORY AND COMBINED NEURAL NETWORKS

Long Short-Term Memory networks have been used in a variety of contexts and use cases. One of which is music, more specifically music computation as a dynamic information system described by (Franklin, 2006). LSTM neural networks are well suited for the task of learning patterns in music and reproduction of songs is that the state of a given song is conceptualized as a complex system that depends on past states. This strength of LSTM neural networks potentially would be useful in the task of modeling demand for public transport as a complex system with current states dependent on past states.

The idea of combining CNN for extracting spatial features with LSTM for taking temporal relationships into account has seen a number of successful attempts at fusing those two methods together in one model for predicting ridership (Zhang et al., 2018) (X. Ma et al., 2019). Inclusion of Parallel CNN-LSTM as well as Augmented CNN-LSTM models in the analysis was inspired by X. Ma et al. (2019). Their research assesses the performance of different prediction methods, including contrasting Sequential and Parallel combinations of the CNN-BiLSTM model. X. Ma et al. (2019) found that a Parallel version of the model performs better than the Sequential counterpart.

X. Ma et al. (2019) use a slightly modified version of LSTM called BiLSTM which stands for Bidirectional Long Short-Term Memory networks - a model that is based on standard LSTM, but it implies passing the input sequence both forward and in reverse. On the conceptual level, this allows the LSTM to be 'aware' of the whole length of the input sequence, instead of just current and past members at any given time. This approach could potentially lead to better results in pattern recognition and network's awareness of the context.

There is also a significant difference in authors' approach towards the CNN portion of the model, specifically in the way that they generate image input for the CNN and the depth of the CNN module of the model. First, X. Ma et al. (2019) attempt to preserve more valid spatial relationship by mapping ridership values onto a matrix and making sure that spatial relationships are preserved on that matrix (relative geographic distance translates into a certain number of empty slots in the resulting image matrix). That way there does not appear to be any information about temporal variation to be available to the CNN module. Lastly, the number of filters that are used in the CNN layers appear very low (16 and 8, or 32 and 16 for another variation).

4.3. DATA

Data access is provided by the Centrum Wiskunde & Informatica (CWI) as part of a research project focusing on building a demand forecasting model for public transport using Machine Learning techniques. The data set contains ridership data aggregated from multiple public transport modes (i.e. metro, bus, train, tram) over the period be-

tween 2017-09-18 and 2019-07-21. Ridership for approximately 60,000 5-digit Origin-Destination pairs is present in the original data set.

CWI and other research institutes participating in the project do not have access to the raw data underlying the data set; due to strict privacy rules, data needed to be aggregated to the point where it does not contain data that could be used to identify individuals. Furthermore, in the original study design, spatial resolution was prioritized over temporal resolution; the data set thus contains information about a large number of Origin-Destination (OD) pairs but has limited granularity on the temporal dimension. OD-pairs are constructed based on 5-digit postcodes of origin and destination of the underlying raw data for individual trips.

For each OD-pair, data is aggregated temporally, which in this case implies that per given quarter there are aggregates for months (e.g. ridership in Jan, Feb etc.), day of the week (e.g. aggregated ridership on Mondays, Tuesdays etc. in Q4-2017), hour of the day (e.g. aggregated ridership for 1-2pm time interval on a workday in Q1-2018). One of the caveats of strict privacy rules is that values for ridership that are lower than 300 are assigned into bins ('1-50', '51-100', '101-200', '201-300'). All the values above 300 are rounded to the nearest 20 (e.g. 1605 will be rounded to 1600, 1631 will be rounded to 1640).

Figure 4.1 shows a visualization of the values. Rows represent OD-pairs (5-digit origin postcode: 5-digit destination postcode), and columns represent daily ridership values for a given OD-pair.

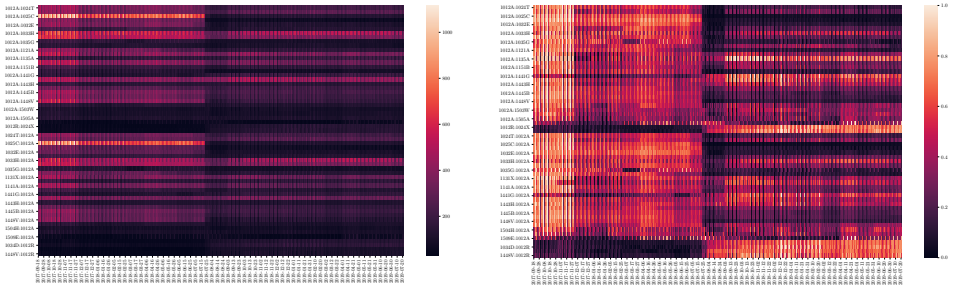


Figure 4.1: (a) Raw input matrix; (b) Scaled input matrix. Rows represent OD-pairs (5-digit origin postcode: 5-digit destination postcode), and columns represent daily ridership values for a given OD-pair.

As can be seen in the raw input matrix in Figure 4.1 (a), there are significant differences between ridership across OD-pairs, with a few having an order of magnitude higher numbers than the rest. To address this, MinMax scaling was used, as implemented in Pedregosa et al. (2011). Each OD-pair (rows) has been scaled independently to ensure that specific demand patterns for a particular route remain intact. The scaled input matrix is depicted in Figure 4.1 (b).

There are a few reasons for using Min-Max scaling on an interval of (0, 1) as opposed to standardization. First is that ML algorithms rely on input being positive and preferably normalized, which speeds up convergence and thus reducing training time. Second reason is that scaling is less sensitive to outliers in the data, since it does not rely on the

mean for a given OD-pair.

4.4. METHODOLOGY

4.4.1. DESCRIPTION OF "IMAGE INPUT" FOR CNN

CNN models are tailored to work with image input data. Normally, images are represented as a matrix of data corresponding to the intensity of color in a given pixel. There is a time-series for each of the OD-pairs that is available, combined in the same matrix (with OD-pair labels on the Y axis and time on the X axis). Data is then mapped to an interval from 0 to 1 using Min-Max scaling technique.

Due to the nature of the convolutions that CNN performs, even spatial relationships that are not bound together should still be identified, however this might require a bigger (deeper) model and/or more time to train. It is thus better to make spatio-temporal patterns more apparent. Thus it is important to note that ordering of the OD-pairs (rows of the resulting matrix) has been adjusted to better reflect the spatial relationships between clusters of trips originating from the same area or having similar destinations. The way that was achieved is through ordering first on the basis of origin, and within the same origin, based on destination. That way, trips that are within the same 4-digit postcode are clustered together as well as trips to and from the same 5-digit postcode are close together. Figure 4.1 (b) contains the visualization of the resulting matrix that is ready to be split into training, validation and test sample sets.

4.4.2. EXAMPLE GENERATION AND ASSEMBLING TRAINING, VALIDATION AND TEST SETS

A sliding time-window of 7 days was used over the whole data matrix. That way a set of examples had been produced that contained 6 column vectors (inputs) in the sample and 1 column vector for the day following that period (expected output). Effectively that corresponds to using 6 days worth of data to predict the following day. This set of examples is then partitioned into training, validation and test sets with 80%, 10% and 10% of the available examples assigned to respective sets. The training set was used for training, the validation set was used in the process of tweaking the CNN model to produce better results, while the test set was used to assess performance.

4.4.3. NEURAL NETWORK MODEL ARCHITECTURE

A summary of model architectures trained using the described example set is presented in Table 4.1. All hyper-parameters have been fixed in order to assess the relative performance between different models. The 'Adam' learning algorithm had been used, with the learning rate of 0.001.

All the models discussed further were evaluated on the same metric, Mean Squared Error (MSE) as calculated in the following formula: $MSE = \frac{1}{n} * \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2$, where n is number of samples in a set, y_i is the actual normalized ridership value to be predicted, $\hat{f}$ is the set of procedures that a given model performs on input data x_i to produce a prediction $\hat{f}(x_i)$. Previous studies such as X. Ma et al. (2017) and X. Ma et al. (2019) have used this loss function for training their models, and it appears to match the context of predicting the continuous variable of ridership.

Table 4.1: Neural network architecture summary.

Model	Input	Architectural choices
CNN	59x6x1	Two Convolutional layers (128, then 64 3 by 3 filters respectively) each followed by Max-Pool layers (2 by 2, with a stride of 2). Same convolutions.
LSTM	1x354	Each example is directly connected to the LSTM layer which contains 354 LSTM neurons that produce a sequence of 354 activation values which are Fully Connected to an output layer with 59 ordinary neurons with ReLu activation function.
Sequential CNN-LSTM	59x6x1	CNN part of that model is exactly the same as in Pure CNN model, however the output of the Max-Pooling layer is flattened into 896 features (14 * 1 * 64), reshaped into (1, 896) to be a valid input for 896 LSTM neurons as a sequence Output of the LSTM layer is consequently Fully Connected to the 59 Neuron output layer as previously.
Parallel CNN-LSTM	59x6x1 and 1x354	Output of exactly the same Pure CNN and Pure LSTM models had been flattened and then concatenated into one vector of 1250 features (896 and 354 from CNN and LSTM respectively) that is then Fully Connected to the output layer with 59 Neurons.
Augmented Sequential CNN-LSTM	59x6x1 and 1x354	Having Pure LSTM connected after CNN, then, may limit the potential of the LSTM layer of capturing time-series features in the sequence. Thus raw input was concatenated to the feature vector produced by the Pure CNN, in an attempt to provide LSTM layer with extra information This augmentation process could also be used to input different kind of data into the model at this stage (e.g. exogenous variables like weather, flags for special events etc.)

4.4.4. ORDINARY LEAST SQUARES REGRESSION

A Fixed Effects Ordinary Least Squares (OLS) model for Panel Data has been used and it is introduced to serve as a baseline for assessing how well Neural Network (NN) models perform. Data from the test period (used for training NN models) was used for estimating OLS coefficients. The same normalization method had been employed as previously. Six lagged ridership sequences were used as input variables to predict the following period. By obtaining OLS predictions for inputs in a test period, MSE could be calculated in the same way as for NN models.

4.5. RESULTS

4.5.1. MEAN SQUARED ERROR AND ACCURACY

Mean Squared Error evaluation values for all models are summarized in Table 4.2. It appears that Sequential CNN-LSTM converges to approximately the same loss value as Augmented Sequential CNN-LSTM over the course of 1250 epochs, and achieves the lowest MSE_{train} on a training set. LSTM, CNN and Parallel CNN-LSTM perform worse, respectively from lower to higher MSE_{train}.

Table 4.2: Evaluation values summary.

	MSE_{train}	MSE_{test}	$\frac{MSE_{train}}{MSE_{test}}$	Accuracy after training
CNN	0.018105	0.010711	0.591604	0.188
LSTM	0.006634	0.006891	1.038736	0.801
Sequential CNN-LSTM	0.000294	0.000934	3.180420	0.833
Parallel CNN-LSTM	0.024177	0.015674	0.648315	0.188
Augmented Sequential CNN-LSTM	0.000263	0.001089	4.148850	0.892
OLS	-	0.010186	-	

Sequential CNN-LSTM, Pure LSTM as well as Augmented Sequential CNN-LSTM appear to reach accuracy of above 80%. Accuracy of both Pure CNN and Parallel CNN-LSTM did not exceed 20% accuracy throughout 1250 epochs of training.

Sequential CNN-LSTM and Augmented Sequential CNN-LSTM models show at least 6-fold reduction in MSE to the second best model (Pure LSTM). On the other end, the performance of both Pure CNN and Parallel CNN-LSTM on a test set is worse than OLS model results (Table 4.2).

The ratio MSE_{test}/MSE_{train} emphasizes the relative difference between training set and test set performance, which might suggest which direction is more productive to follow when improving the model. This will be explored in more detail in the Discussion section.

4.5.2. PREDICTION

Figure 4.2 shows a cross-sectional view of predictions for all OD-pairs for a given day in the test set, and Figure 4.3 shows a time-series view of predictions for the whole test period for a given OD-pair.

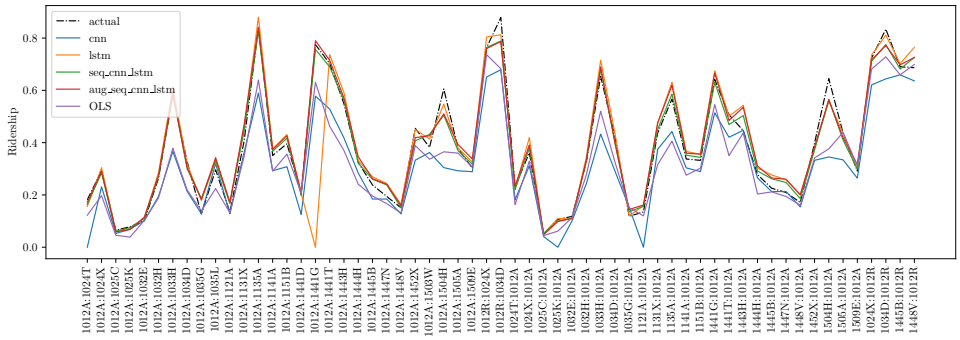


Figure 4.2: Prediction for 2019-06-27 (cross-section view).

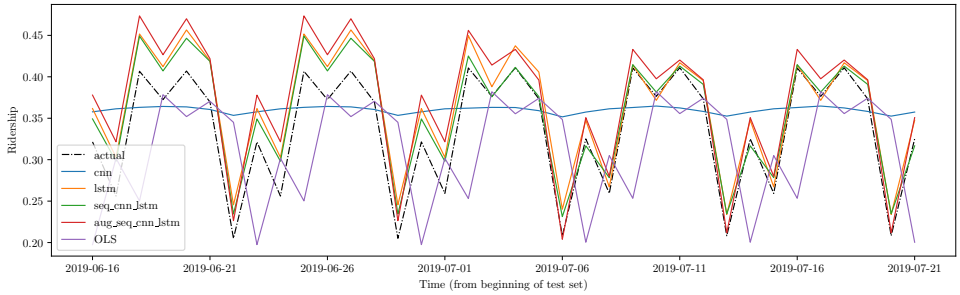


Figure 4.3: Prediction for 1012A:1131X OD-pair, whole test period (time-series view)

In the cross-sectional view in Figure 4.2 it can be seen that models with lower MSE on the test set predict ridership values that are closer to ground truth. Even though it seems like Pure CNN predictions fit the actual values well. Assessing it from time-series view of predictions, as shown in Figure 4.3, for a given OD-pair reveals that CNN, in fact, predicts values in a small proximity to the average ridership of a given OD-pair. This implies that cross-sectional view for Pure CNN performance varies depending how close ridership is to the average for any OD-pair. While predictions of the LSTM model seem to be accurate most of the time, for one particular OD-pair the prediction is consistently null.

4.6. DISCUSSION

It appears that Sequential CNN-LSTM is the best model in terms of performance on the test set. Its score, however, is quite close to that of the suggested Augmented CNN-LSTM model. Moreover, in terms of convergence speed, the Sequential CNN-LSTM model is the fastest in terms of passes (epochs) required to fit the training set.

This suggests that features extracted by the CNN module within the Sequential CNN-LSTM model are sufficient for forecasting ridership accurately. Given that the Pure CNN did not perform as well on its own, it is likely that the LSTM module on top of CNN had played a crucial role in model performance.

While Augmented CNN-LSTM had achieved similar results to its Sequential counterpart, additional raw input does not seem to result in lower MSE. Also, in case the input sequence was larger (e.g. the past 14 days worth of data, or if the time-frame was 10 minutes instead of 1 day), then the concatenated input would require a much larger LSTM layer to be trained, which does not scale linearly in terms of compute time. That said, it appears the augmentation method itself works, and potentially could be utilized to insert exogenous variables or metadata about a given example into the model (e.g. weather data, features of a given example etc.).

Ratios between training and test loss shows that models with better performance have seen a relatively larger ratio of MSE_{test}/MSE_{train} than poorer performing models. Ratio of lower than one implies that Pure CNN and Parallel CNN-LSTM are not able to fit the training well and could potentially be improved by increasing model complexity and/or further hyper-parameter tuning.

Models with exceptional performance, in contrast, seem to over-fit the training set, since MSE_{test} is 3 to 4 times higher than MSE_{train} for Sequential and Augmented CNN-LSTM models respectively. This could be potentially mitigated by including Batch Normalisation for instance in the model, or a technique called "Early Stopping", where performance on a validation set is checked after every epoch and in case of stagnation for a specified number of epochs (or in case the loss on validation set goes up), then training is stopped and the model is prevented from over-fitting the training set.

Interestingly, Parallel CNN-LSTM did not perform as well as expected from the findings of X. Ma et al. (2019). Its performance had been closer to that of Pure CNN. Given that the output layer is a simple Fully Connected layer with 59 neurons, it appears that the fact that there are almost twice the amount of features coming from the CNN part than that of LSTM steers the performance of the model towards that of Pure CNN. It is possible that the results would be better if there had been a deeper NN on top of the Concatenated layer, which could distinguish between CNN and LSTM feature sets and utilize them more efficiently.

The prediction pattern of the OLS model in the time series view (Figure 4.3) reveals that the forecast of the OLS model is equivalent to that of a naive model forecasting the next day to be equal to the last known previous day. However, the OLS performance on the test set is better than both the Pure CNN model and the Parallel CNN-LSTM. This implies that the naive model would be better than those two as well. Nevertheless, it is possible that given more complexity in the input data, Pure CNN and Parallel CNN-LSTM could still outperform OLS considering their theoretical capacity for learning complex non-linear relationships in the data.

Pure CNN does not appear to distinguish temporal relationships in the input data as well as theory would imply, since it is clear that for a given OD-pair CNN predicts ridership value that minimizes the loss function (MSE) for a given OD-pair (Figure 4.3). Cross-sectional view, however, indicates that spatial features are well incorporated in the output of the CNN. It is possible that such outcome is the result of the low resolution of the input, more so in the temporal dimension, since X. Ma et al. (2017) shows that it is indeed possible to train a reliable forecasting Pure CNN model using granular spatio-temporal data.

Visual inspection of predictions suggests that the LSTM model is sophisticated enough

to produce accurate forecasts for the majority of OD-pairs. It is not clear why the OD-PAIR "1012A:1441G" in particular is consistently forecast to have 0 ridership by Pure LSTM. Yet, performance of the Pure LSTM on the other OD-pairs and improvement in performance from Pure CNN to Sequential CNN-LSTM suggests that there is a lot of potential for building accurate forecasting models using multiple data sources feeding into an LSTM layer.

5

USING NEURAL NETS TO PREDICT TRANSPORTATION MODE CHOICE BEHAVIOR

In the Amsterdam metropolitan area, the opening of a new metro line along the north south axis of the city has introduced a significant change in the region's public transportation network. Mode choice analysis can help in assessment of changes in traveler behavior that occurred after the opening of the new metro line. As it is known that artificial neural nets excel at complex classification problems, this chapter aims to investigate an approach where the traveler's transportation mode is predicted through a neural net, trained on choice sets and user specific attributes inferred from the data. The method shows promising results. It is shown that such models perform better when it is asked to predict the choice of mode for trips which take place on the same underlying transportation network as the data with which the model is trained. This difference in performance is observed to be especially high for trips from and to certain areas that were impacted by the introduction of the North South line, indicating possible changes in behavioural patterns, entailing interesting possible directions for further research.

Buijs, R., Koch, T., & Dugundji, E. (2021). Using neural nets to predict transportation mode choice: Amsterdam network change analysis. Journal of Ambient Intelligence and Humanized Computing, 12(1), 121-135.

5.1. INTRODUCTION

In 2018, the region of Amsterdam witnessed the most comprehensive structural change of their public transportation network in more than a century. The opening of a new metro line serving the entire length of the north-south axis of the city has led to rigorous changes in the existing tram and bus network. Analyzing the behaviour of transport movements by individuals is an effective way to assess the impact of a rigorous network change. A standard approach carried out to model transportation behavior is discrete choice analysi, using statistical techniques for parameter estimation. Other approaches involve simulation (Z. Li & Xu, 2019).

This study explores a relatively new method that can contribute to behavioral analysis of transport movements, using a novel data set collected in Amsterdam with an app on smartphones that automatically recognizes activity signals. (Thakur & Biswas, 2020) present a comprehensive survey of smartphone sensor based human activity monitoring and recognition techniques using Machine Learning and Deep Learning. Considering the fact that Artificial Neural Networks are extremely capable of performing well when assigned complex classification tasks (Long, 2019), we consider possible application of this technique within the field of behavioral analysis in transportation.

This chapter is structured as follows: Firstly, a brief literature review is presented, followed by a description of the Amsterdam case study. Next, the data set used and methodology to process it will be discussed. After that, suggestions are made for a neural net implementation to classify mode choice. Finally, results are presented assessing both the proposed methodology and to what extent one might say that behavioral patterns are affected by the network change. Based on these results and the discussion of the proposed methods, recommendations for future research are made.

5.2. BACKGROUND

For several decades, discrete choice modeling has been dominated by statistical models, such as the logit and probit models. This paradigm dates as far back as the 1970s ((McFadden, 1973)) and 1980s (e.g. (Cosslett, 1981)), and is an approach that is built upon in more recent publications (e.g. (Guevara & Ben-Akiva, 2013)).

In 2003, Vythoulkas and Koutsopoulos applied a different approach to this problem. Trying to beat the results obtained by conventional statistical methods, they introduced a neural net structure based on fuzzy set theory to model discrete choice behaviour in transportation. They then applied this algorithm to a small data set that obtained by the Dutch Railways. The data used in this study was collected from surveys and related to transportation mode alternatives in the Dutch city of Nijmegen ((Bradley & Gunn, 1990) and (Morikawa, 1989)). The proposed model performed slightly better in the case study than a logit model constructed for the same purpose. One of the key underlying assumptions in this study was that travelers decide based on simple underlying rules rather then complicated functions $F : X \rightarrow Y$. Those rules were then incorporated into a neural net system.

In the context of Market Share forecasting, a study has been carried out by Agrawal and Schorling in 1996, regarding a comparison between the ANN and multinomial logit method. In brand choice analysis, a hybrid model has been suggested by (Bentz & Merunka,

2000).

Recently, (van Cranenburgh & Alwosheel, 2019) have been among a growing kernel of researchers to again use neural nets in practice in a similar context. In their paper, they describe how an ANN can be trained to investigate decision rule heterogeneity. Their method trains a multinomial classification network to assign users to one of four quintessential decision rules, based on theoretical choice data, where each user was presented a series of choices in order. The results of each user are then combined and fed to the network that classifies the user into one of the four categories. A good overview of papers that have applied Neural Nets and other machine learning (ML) techniques to the problem of transportation mode choice can be found in the literature review by (Hillel et al., 2019).

Currently, one of the particular aims of some of the works that apply specifically Neural Net structures to study human choice or behaviour is to focus on the interpretation of the proposed model. One interesting paper from a different field focused mainly on extracting decision rules from data using a neural net has been presented by (Hayashi et al., 2010). They make use of the *Re-RX* algorithm to extract rules from a pruned neural network. The data set used by the authors contains user characteristics and preferences on eating behaviour, which is also an application of neural nets in a behavioral context.

Although Neural Nets are essentially black box algorithms, it is possible to look beyond merely assessing the predictive power of ML models, using them for the same kinds of analyses that are commonly applied when applying conventional logit models. A study by (S. Wang & Zhao, 2019) focuses on the interpretability of a Deep Neural Net (DNN), proposing a way to numerically compute economical information such as choice probabilities and probability derivatives from the DNN.

Another way of developing a better understanding of the workings of Neural Nets when used in the context of mode choice, trying to introduce some conventional knowledge from the field into the model by modifying the architecture, is proposed by (S. Wang, Mo, et al., 2020). They showed that using a sparse Neural Net architecture, based on underlying assumptions of the Random Utility Mixing (RUM) model, could lead to significantly better results than using a generic fully connected DNN. To get get a better interpretability (S. Wang, Mo, et al., 2020) visualize choice probability functions and compute elasticity coefficients in DNN models using numerical simulations. A special type of Neural Net, called MultiTask Learning Deep Neural Networks (MTLDNNs) is applicable to choice modelling situations where it is useful to combine data from different sources, such as bridging the gap between combining revealed and stated preference data, with the capabilities of automatic feature learning that DNNs possess ((S. Wang, Wang, et al., 2020)).

In earlier research, we introduced a novel way of extracting user-specific features from choice set data and applied this data to a Neural Net model for classifying mode choice, and tested this method on a relatively small subset of an Amsterdam data set (Buijs et al., 2020). In this study, we aim to extend the application of this method to the scope of the entire data set (see sections 5.3 and 5.5 for more context about this data set). We will assess how well our model deals with the changes in the network, and discuss what information regarding the network change can be inferred from our model.

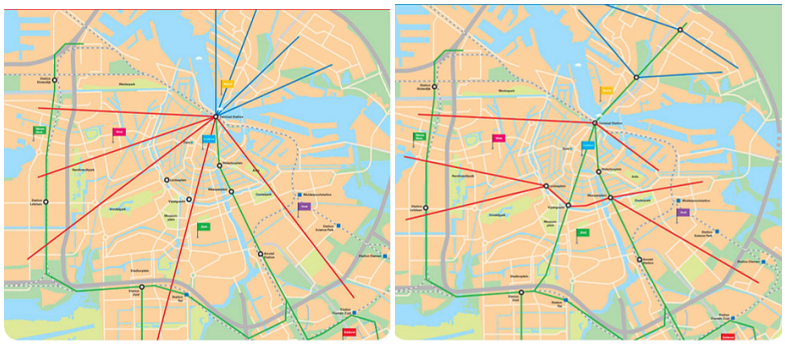


Figure 5.1: The Amsterdam transportation network change visualized by GVB. Conceptually, the upper map shows the old network structure centered around the Central Station, which serves as a hub. The lower map shows the new network structure where the new north-south metro line (the central green line) forms a spine.

5.3. CASE STUDY

In Amsterdam, a new metro line has been opened in 2018 serving the north-south axis of the Dutch capital. In order to improve integration of the new metro line, the existing transportation network underwent significant changes. A large number of the bus and tram lines were re-routed to connect different areas. One particular aim of these changes was to create more east-west links, that connect to the new north-south line at one of the metro stations in the centre of the city. The design moved away from a network that was heavily focused on lines to and from the central train station to a network where instead the new North South metro line forms a spine. [1] An abstract visualization of the network change is shown in Figure 5.1, which was published by the GVB, the main local public transportation provider in the municipality of Amsterdam.[2] For many inhabitants of the city, these changes in the network meant that their personal travel itineraries were affected. At the same time, car drivers were also confronted with the introduction of new restrictions in the inner city and around Amsterdam Central train station to avoid through-traffic in the inner city.

Policy makers from the regional transportation authority in Amsterdam and the city of Amsterdam are keen to assess the impact of the introduction of this new network. For this analysis, data was collected using a smart phone GPS application that was installed by a panel of participants recruited via several existing survey panels. Additional participants were recruited on the street. The smart phone application tracks the activities of the user in the background of the smart phone using sensors on the phone such as GPS and acceleration sensors.

[1] https://www.vervoerregio.nl/pagina/20160131-ov-lijnennetvisie
[2] https://ucarecdn.com/7bad75fd-27ad-4ee6-96c4-b51b2f400619/GVB_Vervoerplan_Amsterdam_2018.pdf

5.4. DATA

5.4.1. CHOICE SET GENERATION

In order to explore what other transportation modes were available for each user for each of their observed choices, we generated a number of alternatives using an open source library developed by Conveyal , R5 - Rapid Realistic Routing on Real-world and Reimagined networks [3]. This router has been used previously by other studies such as (Conway et al., 2017) and (de Freitas et al., 2019). R5 is able to return a large set of feasible, fast routes within a given time-range. This permits a more realistic assessment about accessibility than would be possible using estimations based on fixed frequencies.

We used two separate General Transit Feed Specification (GTFS) [4] data files to feed the router with the correct timetable before and after the opening of the metroline. For the street network we used a temporally appropriate extract from OpenStreetMap. Additionally, we directed R5 to generate transit routes specifically including and excluding metro. For each observation and alternative we then categorized a route into one of 7 different non-overlapping strata:

1. Walk trip (generated if walking stays under 60 minutes)
2. Car trip (generated if destination is reachable by under 60 minutes)
3. Bicycle trip (generated if bicycling stays under 60 minutes)
4. Transit trip, with use of train and metro
5. Transit trip, with use of train (no metro)
6. Transit trip, with use of metro (no train)
7. Transit trip, not using train or metro

To generate choice sets, we looked at the observations and categorized each observation with a stratum and subsequently took the best (fastest) from the alternatives that fit each alternatives. In some cases not each alternative was available, for example walking is not always an option if the distance between origin and destination is long. It could be possible to address the unavailability of walking in the loss function using the study by (S. Wang, Mo, et al., 2020).

5.4.2. FEATURE ENGINEERING

From the observations and the generated alternatives we collected a number of explanatory variables as listed in Table 5.1. We used a walking speed of approximately 5 km/h and a bicycling speed of 14.4 km/h. We based our car speed on the speed limits in OpenStreetMap.

[3] https://github.com/conveyal/r5
[4] http://gtfs.ovapi.nl/nl

Table 5.1: Variables collected for choice set.

Variable	Description
group id	A unique identifier referring to a single trip from origin to destination; generated trips corresponding to the observed trip have the same groupid and also refer to a specific person and date
strata	Categorical variable that indicates the transportation mode of a (generated or actual) trip: 1 for walking 2 for traveling by car 3 for traveling by bicycle 4 for traveling by public transportation with use of metro and train 5 for traveling by public transportation with use of train (no metro) 6 for traveling by public transportation with use of metro (no train) 7 for traveling by public transportation without use of metro and train
access mode	Categorical variable that indicates the mode of access to public transportation (i.e. mode of transportation used to reach the bus stop/train station such as walk, bicycle or car)
egress mode	Categorical variable that indicates the mode of egress from public transportation (i.e. mode of transportation used to reach the destination after leaving the bus stop/train station)
start time	Time that trip started at origin
end time	Time that trip ends at destination
transfers	Number of transfers on the public transportation part of this trip
distance	Total distance of trip
bicycle distance	Total distance of trip traversed on bicycle
car distance	Total distance of trip traversed by car
walk distance	Total distance of trip traversed on foot
bicycle duration	Total duration of trip that is traversed on bicycle
car duration	Total duration of trip that is traversed by car
walk duration	Total duration of trip that is traversed on foot
waiting time	Total time spent waiting on public transportation if applicable

Table 5.2: Amount and fraction of data that were filtered out due to various reasons.

Original data set	**106,647 entries (100.0%)**
No alternatives could be generated	9,894 entries (8.5%)
Trip is a round trip/tour	9,784 entries (9.2%)
Significant parts are traversed by different modes	5,848 entries (5.5%)
Trips that were likely indirect	3,524 entries (3.3%)
Trips that took longer than 100 minutes	885 entries (0.8%)
Final data set	**76,712 entries (71.9%)**

5.4.3. DATA FILTERING

The entire GPS dataset consists of 106,647 trip entries from 712 users. The GPS data is collected during 3 time periods spanning about one month each: The first period of data collection took place in June and July and part of August 2018, largely before the introduction of the new North-South metro line, and the second and third period of data collection took place after the North-South line was opened: in September and October 2018 and June and July 2019, respectively.

Not every trip can eventually be used in a final data set to perform a mode choice analysis on. It was found that several trips in the data turn out to be tours or round trips. As these trips do not consider movement from A to B, these are not suitable to include for a mode choice analysis. Another example of entries that had to be filtered out in some cases, are non-public transport trips, for which different parts of the trip were traversed by a different mode, as these type of trips have not been generated as choice alternatives. These were dealt with as follows:

- Trips with the mode combination of car and cycling are all discarded.
- Trips with mode combination of car and walking are discarded if the duration of the walking part exceeds the duration of the part traversed by car. Otherwise, the trip is considered to be similar enough to a 'car-only' trip, that it is concerned as such, and the duration of the walk is set to 0.
- Trips with mode combination of walking and cycling are discarded if the duration of the walking part is longer than 30% of the duration of the part traversed by bike. Otherwise, the trip is considered to be similar enough to a 'bike-only' trip, that it is concerned as such, and the duration of the walk is set to 0.

For some trips, the difference between the observed trip duration and the theoretical trip duration, determined by the duration of the generated trip with corresponding mode, is rather large, ranging from a factor 2 to a factor 10 difference. These differences may have various reasons, some of which could be a ground for excluding the trips from the data. Trips that manifested such a difference and originally consisted of multiple segments (except for public transport trips), are assumed to be an indirect trip from A to B, thus

would not be of interest, and are filtered out likewise.
In addition to this, it is assumed that an Artificial Neural Net will be able to distinguish between 'real' observed data and generated data if the characteristics of the observed data are too far apart from the range of values that occur in the generated data characteristics. For this reason, all data entries concerning trips spanning more than 100 minutes, are discarded as well. Table 5.2 shows the reasons why some data was discarded and how many entries were involved for each reason. The final dataset consists of 76,712 trip entries, concerning 709 users. For each of these entries, at least one alternative trip has been computed in which a different mode was used.

5.5. METHODOLOGY

This section gives an account of what operations and techniques have been used in order to make mode choice predictions based on the data set. Since we opt for a machine learning approach, most methodological decisions are made such that input is created which is suitable to train a machine learning model on.

5.5.1. DATA PREPARATION

Within the process of data preparation, 3 main steps can be distinguished: Combining data, selecting features, and splitting data into a train, validation and test partition. We will briefly discuss all three of these steps.

COMBINING DATA

Having obtained a filtered data set, the observed data was combined with the data concerning generated alternatives. Initially, duplicates exist in this merged data set, i.e. for a single trip, there can be two routes with the same transportation mode: one that corresponds to the trip that was made originally by the user, the other one is the generated trip having the same transportation mode. In order for the data to be used in a machine learning model, one of the two entries must be deleted, so that for each trip only one option per transportation mode remains. While there are certainly relative advantages to restricting the feature data to one source, most notably that an ML-model will not pick up any bias from the fact that different data sources are combined, it needs to be noted that redeeming the features from the observed trip data means losing valuable information, possibly causing the model to be a worse reflection of reality. Therefore, it is opted for to preserve this data and discard the generated duplicates. Reduction of bias present in the data will be taken into account specifically when selecting features that will serve as input for the ANN.

FEATURE SELECTION

In order to fully benefit from the power of ANNs and to get meaningful results, it is necessary to carefully select the features that will eventually be fed to the ANN. The most important reason for this is to reduce the risk of having 'false predictors' as much as possible. These arise when the Neural Net would be able to distinguish 'real' observed data from generated data within a choice set. From all features initially present in the data, the most reliable predictors will likely be $transfers$, $distance$, $bicycle\_distance$, $car\_distance$, $walk\_distance$, $bicycle\_duration$,

car_duration, *walk_duration* and *waiting_time*. These features together form an initial selection of input features for the ANN. Other features have not been considered as direct input, either due to the nature of the feature or due to the feature having little explanatory value. However, when comparing the generated and observed data, it was found that a substantial number of entries in the generated data had a record of an abnormally high waiting time, resulting in an abnormally high trip duration as well. It is assumed that this is caused by users doing activities not related to transportation at a station. Because of this, it was opted to exclude the feature *waiting_time* from the data and to adjust the feature *duration* accordingly. This operation was performed before the data filtering took place as described in section 5.4.3. It was also observed, that the features related to distance and the features related to duration display different correlation patterns in the observed and generated data. This is illustrated in Figure 5.2. This observation indicates that feeding a choice set including both distance and duration related attributes, might also introduce an unwanted form of bias in the data. Therefore, we choose to discard all attributes related to distance, as it is known that duration plays a more important role in mode choice considerations of individuals.

SPLITTING DATA

A common practice within the field of machine learning is to split the data before it is being used. In supervised learning (training the model to predict a known target), the data is usually split into a train set and a test set. The former is used to train the model, whereas the latter is used to evaluate model performance on a batch of unseen data. From the training set, some data is usually set apart for validation. This part of the training set is not used to train the model, but to check whether the model does not overfit. This would be the case if the model performed significantly worse on the validation set than on the training set. The data for each user is set to follow roughly this distribution over the three sets:

- Training set: 50%
- Validation set: 20%
- Test set: 30%

In order for the model to be able to take into account individual user preference characteristics, it is important that data from all users is contained in the train set. Some users who have only one entry, will as a consequence only appear in the train set. The final sizes of the three partitions are as follows:

- Training set: 38,539 entries (including all single-entry users); 50.2%
- Validation set: 15,156 entries; 19.8%
- Test set: 23,017 entries; 30.0%

The training set is the only set for which the target variable (in this case Strata) is not hidden. Hence, all operations used for setting up the model that are described in the following sections, apply to the training set only.

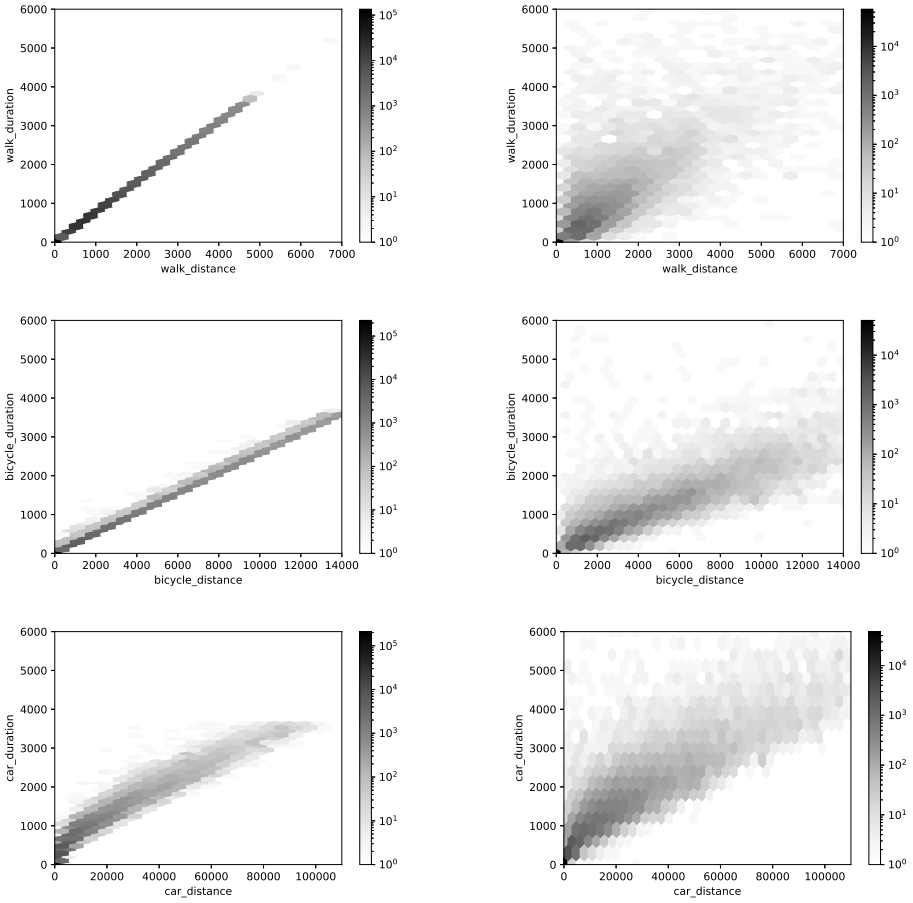


Figure 5.2: Correlation between distance and duration related attributes in the generated data (left) and observed data (right).

5.5.2. CLASSIFYING CHOICES

It is clear that individual users have different preferences. Those individual preferences should be taken into account in any predictive model, as becomes clear in the literature. Because of the nature of the data (panel data), where multiple trip entries will correspond to the same user, it makes sense to include features to the input of the ANN model that specifically concerns individual user preference. The literature suggests different methods in order to classify users as decision-makers, yet all are based on assumptions. The most important notion that comes clear from this is that different individuals decide differently, and can be divided into classes or groups that share similar decision characteristics. Regardless of what the underlying decision functions might be (it will be nearly impossible to approximate them all due to many users having relatively few training entries), it is possible to divide the observed mode choices into different classes based on comparative measures regarding the alternative modes. The comparative measures have been computed by normalizing the attributes *transfers, duration, bicycle_duration, walk_duration* and *car_duration* within each choice set individually and extracting solely the normalized values corresponding to the chosen mode. In this way, for each trip, a singular value between 0 and 1 is obtained for each attribute, where 0 is obtained if the alternative with the lowest value of an attribute is chosen and 1 is obtained if the alternative with the highest value for this attribute is chosen.

K-MEANS CLUSTERING

In order to subdivide the trips into different groups without having a clear target to aim for, a method called k-means clustering is used (Steinley, 2006). K-means clustering is a relatively simple and intuitive clustering method. Although multiple clustering methods exist that can deal with specific types of problems, like Hierarchical clustering (Murtagh & Contreras, 2012) and DBSCAN (Schubert et al., 2017), k-means clustering is fairly suitable for a relatively simple clustering task like the one at hand. With this particular data set, there are two main challenges in terms of clustering:

1. The most obvious underlying structure is the strata classification itself, which tells something about the choice, yet is not the particular information structure we are looking for.

2. Some of the normalized variables may be correlated, for example `walk_duration` and `duration`.

To overcome these obstacles, the following solutions have been suggested:

- Choose the number of clusters k such that k exceeds the number of significantly different modes (In this case, 4: walking, cycling, car, and public transportation) by a comfortable margin (but not higher than necessary) to create substantial 'classes' that are composed of entries from different strata. In this case, k was set to 10.

- Perform principal components analysis prior to performing k-means clustering (Jolliffe & Cadima, 2016). This method creates linearly independent vectors (i.e. vectors that have correlation 0). The resulting vectors are then used as input for the k-means classification algorithm.

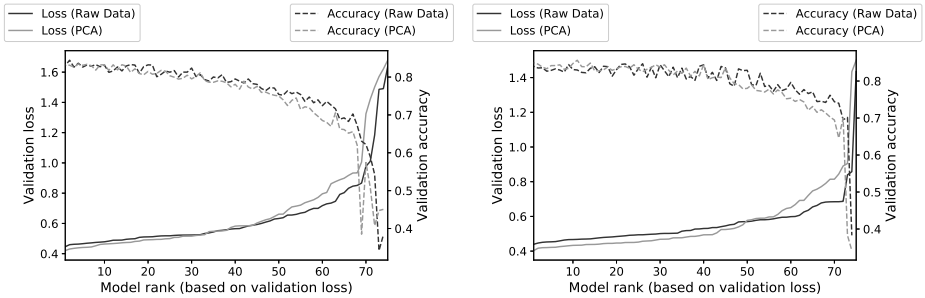


Figure 5.3: Ranked performance of models (in terms of sparse categorical crossentropy loss and prediction accuracy) of ANNs with full user-specific input feature data (i.e. 10 distinct features) and input where user-specific features are reduced in dimension (i.e. 5 distinct features). In the upper plot, early stopping is applied when validation loss has not decreased for 2 consecutive epochs, whereas the lower plot shows models that were trained with the application of early stopping condition when validation loss has not decreased for 5 consecutive epochs.

5

PRINCIPAL COMPONENT ANALYSIS

Our next step is to gather information about users using the obtained choice classification. Based on the outcomes of the labeling phase described earlier, each user now has a characteristic 'label distribution'. The relative frequencies of the different choice types are stored in a DataFrame for each user. Again, PCA is conducted to reduce these values to a set of 5 vectors that aim to capture the users' behavior and taste.

In order to assess the usefulness of applying PCA here, we trained 75 models with hyperparameters randomly selected from a the hyperparameter space as described in Table 5.4, with PCA applied to the user specific features and ranked the models based on validation loss. After that, the same procedure was applied to assess the performance of models that were trained based on input where PCA was not applied to the user specific features. This was done once applying an early stopping condition of 2 epochs, and then repeated once more applying an early stopping condition of 5 epochs.

The results of this procedure are shown in figure is assessed in Figure 5.3. The figure indicates that for the top-ranked models, models where PCA was applied to the user specific input features have a slightly lower loss than models where this was not the case. The reported accuracy values of the highest ranked models are roughly the same for both the models where PCA was applied to the input and the models where PCA was not applied.

5.5.3. PREDICTION

In order to predict which strata will be chosen in different situations for different users, we feed the acquired data concerning trips made, alternatives, and user preference to an artificial neural network (ANN).

NEURAL NETWORKS FOR MULTICLASS CLASSIFICATION

Neural nets have extended the scope of machine learning beyond linear models. A feed-forward neural network consists of one or more hidden layers, that each consist of a

number of nodes. In a basic, fully connected neural net, each node gets input from all nodes in the previous layer, and outputs to all nodes in the next layer. Each layer is assigned a type of activation function, i.e. a function that generates the output of a node based on the input from a previous node. Commonly used non-linear activation functions include *sigmoid, tanh* and *reLU* functions, with respective domains (0,1), (-1,1) and [0,∞). All nodes except for those in the output layer have $reLU$ as activation function, which is the most commonly used activation function nowadays. For a multiclass classification, the method that will be used in this study, another activation function is usually used in the final (output) layer. The so-called *softmax* activation takes exponents of the output of the previous layer and scales them such that they sum to 1. The network is trained by a back-propagation algorithm that works upon a chosen loss function. Commonly used loss functions include least-squares and cross-entropy loss. The network is trained with *rate* η. After each iteration the weights are adjusted in the direction of the gradient of the chosen loss function, based local derivatives and the chain rule. The *training rate* η is the parameter determining the magnitude of the change of weights after each iteration. Choosing a higher value for η increases the training speed but may result in not being able to find optimal values. It is possible to train a Neural Net with a constant value for η for all parameters, or with adaptive η, meaning that η can be different for each parameter update. Different optimizers have been introduced that make use of adaptive η, like Adagrad (Duchi et al., 2011), RMSProp (Tieleman & Hinton, 2012) and Adam. In this study, we use Adam as optimizer, since it is "robust and well-suited to a wide range of non-convex optimization problems in the field machine learning" and the method is computationally efficient, which is convenient for large data sets (Kingma & Ba, 2014).

DATA SHAPE AND PROCESSING

In this case, the shape of the individual data entries fed to the network is a table of 7 by 10; 10 attribute values for each of the 7 different alternative strata. Each *groupid* in the training set corresponds to one mode choice scenario and therefore to one of these tables. The values of the 10 attributes are not always available for every stratum as not every mode of transportation is possible on every trajectory. If no route was generated for a certain stratum in a certain scenario, all attributes corresponding to this stratum (including user-specific attributes that are essentially known even for alternatives that do not have a route generated) are set to 0 in this scenario, and will be passed to the network as such. This is done because the network requires the data entries passed to it to have a consistent shape (*Stratum 1= row 1, Stratum 2 = row 2 etc*), while in the meantime strata without a generated route option must not affect the working of the model. Before creating the data entries, all data has been normalized using the minimum and maximum values of the entire combined dataset.

5.6. RESULTS

5.6.1. CLASSIFYING CHOICES

Table 5.3 shows the composition[5] of the clusters that result from the k-means clustering algorithm with $k = 10$. After inspection of the clusters, a description has been added based on the values of the used comparative measures observed among the choices in each cluster.

5.6.2. PREDICTION

For this chapter, we trained 75 ANNs with different hyperparameter configurations. All models are trained for a maximum of 100 epochs, where early stopping is applied if the loss on the validation set does not decrease for 2 consecutive epochs, to prevent the model from overfitting on the training data. The hyperparameters concerning architecture were randomly chosen for each run from a pre-determined set of possible values, as given in Table 5.4. The models are trained using the Adam optimizer which was mentioned in section 5.5.3, with fixed α, β_1, β_2 and ϵ (Table 5.4. Based on the validation loss, the best 5 models were selected, as can be seen in Table 5.5. For these models, we looked into the confusion matrices for the classifications. Table 5.6 shows the confusion matrix for the highest ranked model, based on the predictions made on the test set. As we see, the model performs very well on choice sets where the actual mode was car. Decent scores are also reported for all other Strata, except for Stratum 4. This is likely due to the fact that this is the smallest class containing only 96 entries in the training set.

5.6.3. ANALYSIS IN THE LIGHT OF THE NETWORK CHANGE

In addition to the assessment of our model in general, we have explored the effect of training our model on the different partitions of the collected data. For this, we compared two different settings: One where the entire model was trained and validated based only on data relating to trips made before the introduction of the North-South line (this is data stemming from the first collection period), one where the model was trained and validated based only on data relating to trips made after the North South line was introduced (data from the second collection period). These models were than tested on data from the final collection period. As a reference, we also trained a model with roughly the same amount of data (about 20,000 entries) sampled randomly from all three collection periods and tested this against a test set containing a random sample of the remaining data. For all of these settings, 10 models were run with 4 hidden layers, 200 hidden nodes per layer and batch size 128. This approach is mainly used to obtain a relative insight into how much the classification task will become 'different' when the underlying network is different. If the assumption that the data on which the model is trained and the data on which the model is tested are i.i.d. is not valid, the model will not be able to perform as well on the test data in comparison to situations where this assumption does hold. This principle is related to the theory underlying transfer learning.

[5] When running the model multiple times, the exact composition of the clusters will slightly deviate due to the nature of the clustering algorithm. The nature and sizes of the identified clusters however have shown to be consistent over multiple runs.

Table 5.3: Composition and description of clusters obtained by k-means clustering after applying PCA.

	Stratum							
Cluster	1	2	3	4	5	6	7	Total
0	1,270	0	0	0	5	6	3	1,284
1	0	13,412	0	0	0	0	0	13,412
2	0	0	9,994	0	0	0	0	9,994
3	4,273	0	0	0	26	5	34	4,338
4	0	0	0	34	558	353	372	1,317
5	0	0	3,472	0	0	0	0	3,472
6	0	707	0	0	0	0	0	707
7	0	0	0	2	269	88	578	937
8	0	0	0	60	324	404	548	1,336
9	273	0	0	0	302	264	903	1,742
Total	5,816	14,119	13,466	96	1,484	1,120	2,438	38,559

Cluster	Size	Description
0	1,284	Trips involving walking which are not the most time-consuming alternative
1	13,412	Trips by car which are generally (among) the quickest alternative(s)
2	9,994	Trips by bicycle which are generally among the quickest alternatives
3	4,338	Trips where relatively more walking is involved, generally the slowest
4	1,317	Public transportation trips with relatively many transfers and (relatively) longer walk involved
5	3,472	Trips by bicycle which are generally among the slowest alternatives
6	707	Trips by car which are generally among the slowest alternatives
7	937	Relatively slow public transportation trips with relatively few transfers
8	1,336	Public transportation trips with relatively many transfers with relatively fewer walking
9	1,742	Relatively fast trips without using car or bicycle or making a lot of transfers

Table 5.4: Hyperparamter configuration for the ANN.

Hyperparameters: randomly sampled for each run	
Hyperparameter	Values to sample from
Number of hidden layers (excluding output layer)	{1, 2, 4, 6, 8, 10}
Number of nodes in each hidden layer	{5, 10, 25, 100, 200, 500, 1000, 2000}
Batch size	{8, 32, 64, 128, 512, 2048, 8192, 38539}
Hyperparameters: Fixed for every run	
Activation function for every layer except output layer	*reLU*
Activation function output layer	*Softmax*
Number of nodes output layer	7
Loss function	Sparse categorical Cross-Entropy Loss
Learning rate α	0.001
β_1	0.9
β_2	0.999
ϵ	10^-7

Table 5.5: Results and characteristics of the 5 best models selected on validation loss.

Rank	Number of hidden layers	Number of hidden nodes	Batch size	Number of epochs trained	Validation loss	Validation accuracy
1	4	200	128	17	0.422	0.843
2	4	200	64	13	0.433	0.834
3	4	1000	512	11	0.438	0.835
4	4	100	128	16	0.440	0.834
5	6	100	32	16	0.440	0.835

Table 5.6: The confusion matrix of the highest ranked model.

	Predicted Stratum							
	1	2	3	4	5	6	7	Predicted correctly
Actual Stratum								
1	2,883	148	490	0	2	6	7	81.5%
2	164	7,866	635	2	65	97	40	88.7%
3	744	714	6,323	0	6	69	33	80.1%
4	0	66	2	25	8	5	2	23.1%
5	2	190	31	0	697	3	0	75.5%
6	1	44	39	0	5	492	5	84.0%
7	9	145	97	0	1	23	831	75.1%

An interesting further research direction, yet outside the scope of this chapter, would be to try to further generalize our model using the transfer learning techniques discussed by (Yosinski et al., 2014)), by subjecting our model to two similar waves of data with a different underlying network.

Table 5.7 shows the differences in (relative) cluster sizes for the clusters that were obtained based on the 2 different collection periods. We can see that most clusters formed are very comparable in composition and size for the two periods. However, it is worth noting that the share of fast public transportation trips with relatively less transfers has dropped, and the share of public transportation trips with relatively many transfers has risen. This is in line with what one would expect given the new network structure with the North-South line as a spine.

Figure 5.4 shows that the models that were trained on data from the second period of data collection, generally had a higher performance on the test set (data from the final collection period) than models that were trained on data from the first period of data collection. This implies that the Neural Net is better able to capture mode choice relations if the underlying transportation network is the same. Both groups of models however generally performed worse than those trained and tested on data that was randomly sampled throughout the entire data set. It was also found that the difference in performance between the models trained data from the first and second collection period depends on the origin and destination of the trips in the test set. If we compare, for example, the subsets of trips that had an origin in one neighbourhood containing a North South line station and destination (*North* - Noord, *City Centre* - Centrum and *South* - Zuid) in another neighbourhood containing such a station, we mainly observe an increase in prediction accuracy for the second group of models compared to the first when looking at trips going from Centrum to Noord (see Figure 5.5). The differences in accuracy between the two model categories in Figure 5.5 may be an indication of the similarity between the classification tasks with differing underlying networks for trips between these areas. Especially, the remarkable difference observed for the Centrum-Noord trips might indicate

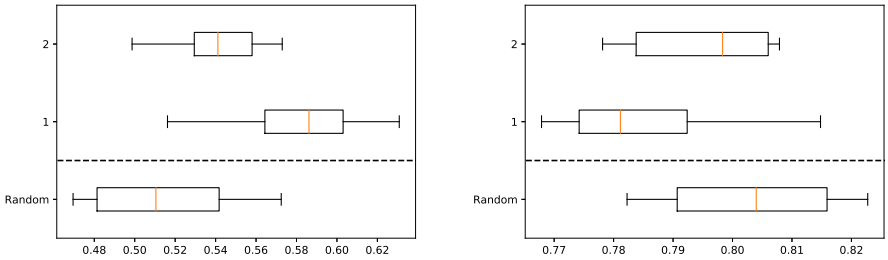


Figure 5.4: The loss (left) and accuracy (right) of models trained on data collected in the first and second period, when tested against data collected in the final period, compared to the loss and accuracy of a model with randomly sampled train and test set of roughly equal size.

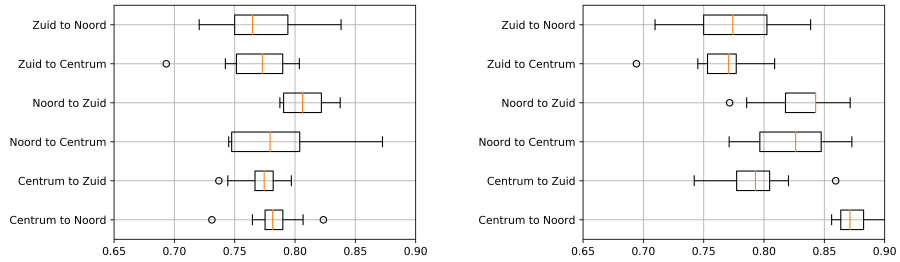


Figure 5.5: The accuracy of predictions of trips grouped by the neighbourhoods of origin and destination, for models trained on the first (left) and second (right) data collection period.

that the underlying behavioral patterns relating to travel between these areas are to some degree different prior to and after the network change.

5.7. DISCUSSION

As tables 5.5 and 5.6 suggest, using a multi-layer ANN with sufficiently many nodes in each hidden layer can be a useful and promising technique in predicting mode choice for a large GPS dataset. As mentioned earlier, one of the disadvantages of using a fully connected Neural Net in order to predict mode choice based on multiple sources of data, is that the Neural Net will likely pick up on any pattern that is related to the source of the data, which can cause the model to learn non-meaningful relations. In order to investigate which are predictors the model most heavily relies on, we also trained models where several attributes were excluded from the Neural Net input (feature selection by elimination). We tested 6 different settings that are relevant for the assessment of our model, for which 10 models were trained each. In 5 of the 6 settings, a specific attribute or set of attributes was removed from the input data, and in one setting, no attributes were removed. All models were trained using 4 hidden layers with $reLU$ activation and 200 hidden nodes per layer, with batch size 128. The performance of these models in

Table 5.7: Comparison of cluster size and composition between the first and second data collection periods.

Relative size 1st period	Description	Relative size 2nd period
34.2%	Trips by car which are generally (among) the quickest alternative(s)	36.9%
13.1%	Walking trips or public transportation trips where relatively more walking is involved, generally the slowest	10.4%
3.1%	Relatively slow public transportation trips with relatively many transfers and (relatively) longer walk involved	3.1%
1.8%	Trips by car which are generally among the slowest alternatives	1.8%
2.4%	Relatively slow public transportation trips with relatively few transfers	2.2%
5.9%	Relatively fast walking trips and relatively fast public transportation trips with relatively less transfers	4.4%
3.2%	Public transportation trips with relatively many transfers and relatively less walking	4.1%
6.7%	Trips by bicycle that are generally among the slowest alternatives	8.2%
19.9%	Trips by bicycle that are generally among the fastest alternatives	25.4%
9.8%	Trips by bicycle that are generally among the slowest nor the fastest alternatives	
	Walking trips in general, that went faster than a generated public transportation trip with a great walking component	3.5%

Table 5.8: 10%-trimmed mean of loss and accuracy of model predictions on the test set when certain attributes are excluded during training.

Excluded attributes	Test set loss (10% trimmed mean)	Test set accuracy (10% trimmed mean)
None	0.513	0.807
User-specific attributes	0.540	0.782
walk_duration, car_duration, bicycle_duration	0.602	0.776
transfers	0.504	0.810
duration	0.522	0.801
User-specific attributes, *transfers, duration*	0.721	0.727

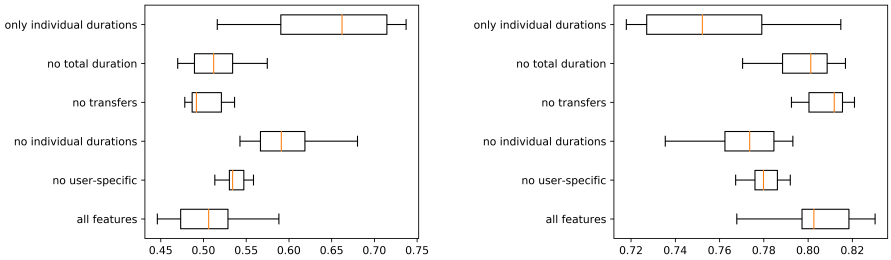


Figure 5.6: The test loss (left) and accuracy (right) of models trained with different attributes excluded during training.

terms of the losses on the test set and accuracy are displayed in Table 5.8 and Figure 5.6. The results suggest that the ANN is generally less prone to overfitting on the train set if the attribute *transfers* is excluded. The variation in performance can also be reduced by removing the user-specific attributes, however the high losses that are obtained can partially by the fact that the training process is stopped when the validation loss does not decrease for a period of 2 consecutive epochs, while in fact the validation loss could likely decrease more if the model training would not have stopped early. Excluding *car_duration*, *bicycle_duration* and *walk_duration* leads to a greater reduction in model performance. This fact may suggest that these are important predictors, hinting at a possible relation between these variables that would lead the Neural Net to detect which trip was a real record (and not a generated one). However, the models based on only these predictors perform significantly worse than all other tested settings, which weakens this assumption somewhat.

5.8. CONCLUSION

This chapter examined the usage of ANNs in order to predict transportation mode choice using a combination of a large GPS-based data set and additionally generated data. After combining and filtering the data, extra user-defining features were extracted using k-means clustering and PCA. Several ANN models were trained based on the choice sets and these extracted features, with different (randomly sampled) hyper-parameter settings. The best model initially reported a validation accuracy of over 84% and performed well in predicting trips from every category, except for trips from the smallest one (public transportation with use of train and metro). It was found that the model performed better on unseen data if the data on which it was trained and tested, were collected on a very similar underlying transportation network, than if the underlying transportation network would be somewhat different between the train and test sets. This difference in performance was observed to differ based on the origin and destination area of the trip for which the mode had to be predicted. After further analysis, it was found that excluding durations of each individual mode from the training data has the highest negative impact on model performance, whereas excluding the number of transfers has little to no negative impact, and might even reduce overfitting.

Building on this study, interesting for future research might be to investigate a more problem-tailored Neural Net architecture, as well as to infer all information that could be inferred from classical statistical mode choice models from the ANN.

As the source data for this neural network is panel data, it's possible that better results could be achieved by a model that is not blind to panel effects. The work by (Y. Yang et al., 2020) proposes a new class of interpretable neural network models achieving both high prediction accuracy and interpretability in regression problems with time series cross-sectional data, might improve the accuracy achieve by this model.

Due to the methodological focus of this chapter, the selection of input features for the choice set has been limited to some extent. Another interesting suggestion for future research would be to investigate how well the presented models are suited when the input feature space is extended beyond core attributes like duration. Additional features that could be considered for inclusion would be economic features like parking tariffs, fuel cost or public transportation cost. Also, it would be interesting to see how the model performs when external features are added to the input, such as weather (which is easily extracted from the original trip record database) or trip purpose (which, although not readily available, could be extracted using an activity detection algorithm, see e.g. (Reumers et al., 2013)).

Since the results also seem to hint at a clear change in behavioral patterns following the opening of the new metro line and the restructuring of the network, it would certainly be interesting to investigate these in more detail. Given this change in behavioral patterns, studying and tuning a tailored transfer learning model to this data is another, final possible direction for further research.

III

Part III: Travel Time and Congestion

6

THE OPTIMIZATION OF TRAFFIC COUNTER LOCATIONS IN MULTI-MODAL TRANSPORTATION NETWORKS

In this chapter we will investigate ways to optimize the placement and number of traffic counters used in multi-modal transportation analysis studies for motorized vehicles, bicycles and pedestrians. The goal is to strike a balance between using as few as possible traffic counters for economical efficiency and deploying more counters which could collect more data. By using shortest path algorithms to determine the paths between the centroids of statistical divisions, we derive from origin-destination matrices which traffic is flowing from where to where over which links in a multi-modal network. Using centrality measures such as betweenness, we determine the links in the transportation networks that capture the most useful traffic in terms of as much unique traffic as possible. Next we look at ways to implement additional criteria in the selection of locations: those that are permanently covered, locations that were used for previous studies in prior years for which historical analyses can be made, and locations that capture more than one modality for vehicles, bicycles and pedestrians. Finally, we study groups of traffic counters organized in screen lines.

Koch, T., van der Mei, R.D., & Dugundji, E. (2018). The optimization of traffic count locations in multi-modal networks. Procedia Computer Science, 130, 287-293.

6.1. INTRODUCTION

For the city of Amsterdam, we have been investigating methods how to optimize the planning of where to put traffic counters that measure the traffic circulating within the jurisdiction of the city. In prior years, the planning for the placement and number of traffic counters was determined manually based on intuition and operational experience; the city was interested if a data-driven approach could lead to improved deployment of the counters.

An optimal planning for traffic measurement is a balance between deploying as few counters as possible while not significantly decreasing the coverage of travel movements within the city. For example we might prefer to catch less traffic with more measurement points because we prefer to organize the counters within screen-lines. With screen-line analysis all movements with an origin of travel at one side of a screen line and the destination on the other side are intercepted H. Yang et al. (2006). The benefits of screen-line analysis is that it enables a comparison between the results of traffic assignment and traffic count data along a screen-line; by comparing the sum of traffic volumes in the assignment with the traffic counts, the ratio can be computed "What is Screenline Analysis?" (n.d.). Other reasons to prefer certain locations could be policy reasons such as a high number of traffic complaints in a certain area, future developments in a region or certain spots that are already often permanently covered by counters such as tunnels, bridges and intercity highways.

To conduct the research we had access to multiple sources of data, such as origin-destination matrices sourced from Google and the city council, the measurements of the traffic counters in previous years and speed measurements sourced from Google and city traffic control cameras. To model the street network we used geographical material from the national government (NWB), OpenStreetMaps and the GIS data used within the official traffic model of the city of Amsterdam.

6.2. BACKGROUND

A number of studies have been done in this area. H. Yang and Zhou (1998) derived four rules to locate traffic counting points based on the theory of maximal possible relative error in origin-destination (O-D) matrix estimation: 1) the O-D covering rule, 2) the maximal flow fraction rule, 3) the maximal flow-intercepting rule, and 4) the link independence rule. The theory of maximal possible relative error "represents the maximum possible relative deviation of the estimated O-D matrix from the true one or the upper bound of the real relative error for a particular fitted O-D matrix". The O-D covering rule states that traffic counters should observe at least a certain portion of traffic between O-D pairs at at least one counter. The maximal flow fraction rule states that for an O-D pair the traffic counters should be placed on links where we find the largest flow fraction between that O-D pair and all flows on that link. The maximum flow-intercepting rule states that under a minimum number of links to be observed, these links should intercept the highest amount of traffic flow possible. The link independence rule states that the traffic counters should be located on the network where the resultant traffic counts on the chosen links are not linearly dependent. H. Yang and Zhou (1998) formulate the problem of locating traffic counters as a mathematical problem, where the O-D covering

rule and link independence rule are applied as constraints in the attempt to maximize the total traffic flow observed, solved with a heuristic greedy algorithm.

Chootinan et al. (2005) published a distance-based genetic algorithm. They investigated two methods to solve the problem: a weighed-sum method and a distance-based method. With the weighted-sum method they combine two objectives, covering as much traffic as possible while having the lowest number of counting points, into a single objective. By adjusting the weights different solutions will be found, switching between covering more traffic or having more points. In the distance-based method their main idea is to evolve the genetic search to get closer to Pareto optimal solutions. Their results indicate that the distance-based method is able to provide a better description of the quality-cost trade-off than the weighted-sum method. Additionally it can generate non-dominant solutions in the duality gap that can not be represented in a weighted-based search.

Ehlert et al. (2006) built a software application based on the heuristics published by Yang and Zhou H. Yang and Zhou (1998) and applied it on the network in the district of Gateshead in Northeast England, with 1980 O-D pairs on a network of 240 kilometers of classified roads in 1414 directed links within an area of approximately 142 square kilometers. Additionally Ehlert et al. (2006) proposed two extensions to take budget restrictions into account. Their first extension takes into account locations of existing detectors by using the second-best formulation of the optimization problem and then using these locations to define new link choice proportions. Their second extension prioritizes O-D pairs based on the average information content of an O-D movement taken from previously collected O-D matrices.

C. Yang and Liu (2009) proposed an enhanced genetic algorithm to solve two traffic counting location problems. Firstly they address the problem of determining the optimal number of locations of traffic counts that will cover all the O-D pairs in the network. Secondly they determine the maximum number of covered O-D pairs with a defined number of locations. The enhancement of the genetic algorithm is due to the selection, mutation and partial elitism policy leading to better and faster results.

A. Chen et al. (2007) developed strategies to select additional locations beyond an initial set of already existing traffic counters in order to improve origin-destination trip table estimation. Using these counts the O-D trip table is estimated using a modified flow path estimator that is capable of internally consistent handling of the traffic count. To solve this NP-hard combinatorial problem, they developed another genetic algorithm embedded within a shortest path algorithm.

Barceló et al. (2012) proposes a modified set that formulates the link detection layout with side constraints. Additionally it presents a new meta-heuristic tabu search algorithm with a high computation efficiency. Their solution focuses on sensors that can follow vehicles using the electronic signatures of phones with bluetooth, allowing the placement of sensors at intersections. Specifically Barceló et al. (2012) proposed a new formulation in terms of a node covering problem with side constraints that can be efficiently solved by solving software.

6.3. NETWORK

The road network of Amsterdam within the boundary of the city contains 18,438 links. The VMA ('Verkeersmodel Amsterdam' / Amsterdam Traffic Model) has 4,418 traffic analysis zones that represent origin and destinations for journeys between Amsterdam and any region up to neighboring countries. The city itself has a ring road A10 and a radial secondary network in the form of S-roads that connect the neighbors to and from the ring-road. A water body (the 'IJ') separates North and South districts of the city, which are only connected via two tunnels on the A10 ring road and two S-road tunnels.

To create a graph to represent the network, you would ideally use geographical material with only one edge between each intersection as multiple edges would dilute the intensity of the traffic. Geographic material from OpenStreetMap often uses 3 separate edges to represent one street with cycle-lanes, motorized vehicle lanes and a tramway in the middle, making it a poor choice to use in our computation.

6.4. CALCULATING PATHS BETWEEN AREAS

To gain insight which and how much traffic is flowing from and to where, we calculated the fastest paths between each origin and destination (O-D) pair. To do that we used the open source routing library pgRouting to calculate all the links that are on the fastest path using the A* algorithm. As the cost-function we used the travel-time for each direction on the link.

For each link we store a tuple $(ID_{origin}, ID_{destination}, ID_{edge}, \text{forward}, \text{backward})$, denoting the identifiers of the origin and destination, the link number and whether the path is following the direction of the 'LineString' geometry on the link.

The full collection of these tuples represents an all-or-nothing assignment than can be used to distribute O-D matrices along the network. Using this collection and an O-D matrix it is possible to determine per link how much traffic flows along both directions of the link. Additionally it is possible to calculate the traffic between a subset of the origin-destination pairs along a link, for example to exclude traffic already captured along other links.

6.4.1. SORTING LINKS BY AMOUNT OF TRAFFIC

Using the traffic assignment and an O-D matrix it is possible to sort links by the amount of traffic flowing over them. This is a centrality measure called betweenness, the ratio of how often a vertex or link is part of a shortest path between one of the other nodes in the graph. This is given by $\sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$, where σ_{st} references the number of shortest paths from s to t , and $\sigma_{st}(v)$ the number of shortest paths from s to t that pass through v.

To avoid catching the same traffic at the links more than once, only traffic between O-D pairs not covered yet is counted. To do this it is necessary to keep a record of which O-D pairs are already covered.
By using the PostgreSQL database server, we keep the table with the route assignment and a table with the O-D matrix. To query the link with the highest number of journeys, we join these two tables, and pick the link with the highest aggregated count journeys per edge. To exclude traffic already captured in this query, we keep a third table with origin-destination pairs already covered. The moment we pick a link, we insert all

origin-destinations that travel over that link. We continue this process until all origin-destination pairs are separated by at least one link.

The process can be customized, for any reason such as links that are permanently covered or areas that require additional scrutiny. By initializing the process with predefined links, the algorithm will seek an optimal solution with the remaining links.

6.4.2. SORTING SCREEN-LINES BY AMOUNT OF TRAFFIC

An alternative strategy instead of placing traffic counters on links to capture the highest share of traffic might deliver more useful data. For example it can also be worthwhile to organize the counters within screen-lines. Screen-lines are groups of segments separating traffic between the origin and the destination. Screen-lines can help the validation of Origin-Destination matrix estimations using traffic counts. In our experiment we used previously determined screen-lines that were used in earlier transportation studies by the city. Using the same screen-lines allowed for a better comparison with the data collected in previous years.

To sort screen-lines by the amount of unique traffic we can apply a similar process. For each screen-line, we sum the weight for each O-D pair that covers more than one link that is part of the screen-line. To exclude traffic already covered, we exclude origin-destination pairs separated by at least one link.

In order to weight screen-lines, we use two criteria: the amount of traffic covered and the number of counters within a screen-lines. With a simple weighting function: $intensity/n_{points}$ we observed a disproportionate avoidance of screen-lines with more traffic-counters. As a screen line with more counters results in more useful data than a screen-line with only two counters, we reduced the denominating with the eighth-root of the number of counters: $intensity/\sqrt[8]{n_{points}}$. This weighting function was used to pick the best screen-line each time and then look at subsequent screen-lines based on traffic not yet covered.

6.4.3. CYCLING AND WALKING MODALITIES

Besides cars and other vehicle traffic, the city of Amsterdam also deploys sensors to count the number of cyclists and pedestrians. Since the city itself has no complete O-D matrix for these modalities we had to use different sources. For cyclists we had access to data collected by cyclists who volunteered to log their activities for a week with an application that uses the GPS and sensors on smart-phones. Additionally we had access to an O-D matrix supplied by Google, which contained the movements between areas of pedestrians, cyclists and vehicle-passengers in June of 2016. As geographical material we used OpenStreetMap which has a complete coverage for the cycleways in Amsterdam.

The difference between fast-traffic and slow traffic is that journeys are shorter on average and that there are much more possible paths. This results in less prominent locations to place counters. Since there is an economical advantage to also use the counting locations for bicycle where possible, we used the top-100 locations as the starting point for this computation. Each segment allowing bicyclists parallel to the location covered by a car counting location was selected as a counting location for bicyclists. The remaining locations were then primarily segments that only allowed slow traffic.

Table 6.1: Amsterdam screen-lines sorted by unique traffic coverage

screenline	Journeys covered	$n_{counters}$	$Coverage/n_{point}$	$Coverage/\sqrt[3]{n_{point}}$
Zuidas	169247647	4	42311911	142319739.652
Zuidoost	158309305	6	26384884	126542828.021
Amstel	135445001	5	27089000	110762240.020
Riekerpolder	113804400	2	56902200	104359094.934
Schinkel	104172820	4	26043205	87598550.905
Noord-kanaal-oost	99599079	4	24899769	83752508.494
Amstelveen	86381721	5	17276344	70639985.560
Westpoort/Haarlemmerweg	76489739	4	19122434	64319947.329
West	72324022	4	18081005	60817010.837
Singel	82659889	13	6358453	59986275.630
Noord-vanaf ring noordwest	54713890	3	18237963	47693306.907
Vondelpark	31482064	3	10494021	27442460.049
Gooiseweg	27784370	2	13892185	25478379.628
West(noord/zuid)	21690791	2	10845395	19890543.047
Boerenwetering	23159533	5	4631906	18939065.554
Centrum	15916517	6	2652752	12722695.444
Stadhouderskade	4391555	5	878311	3591261.880

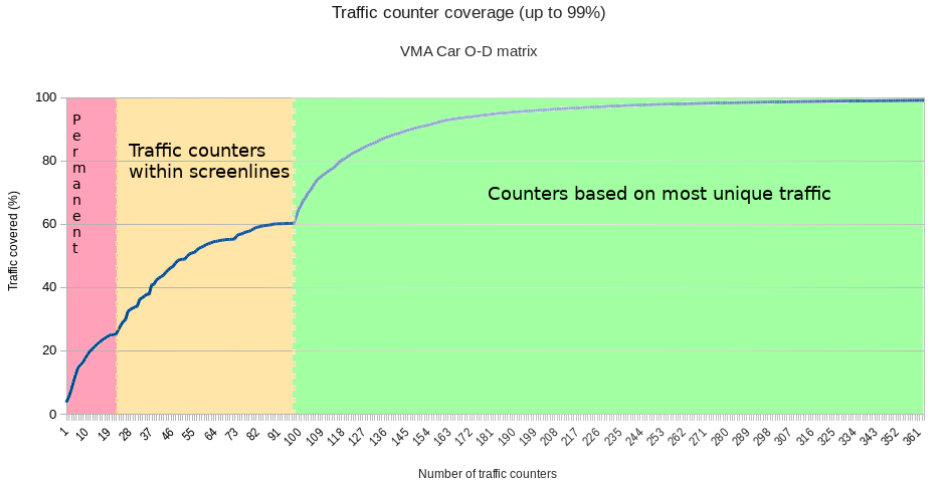


Figure 6.1: Chart of unique traffic of cars covered. The first section includes coverage of all permanently covered locations and the second section is grouped within screen-lines. Note the step-like behavior. Each counter within each screenline is included in the order of the most unique traffic per counter per screenline. The third section contains the most remaining unique traffic.

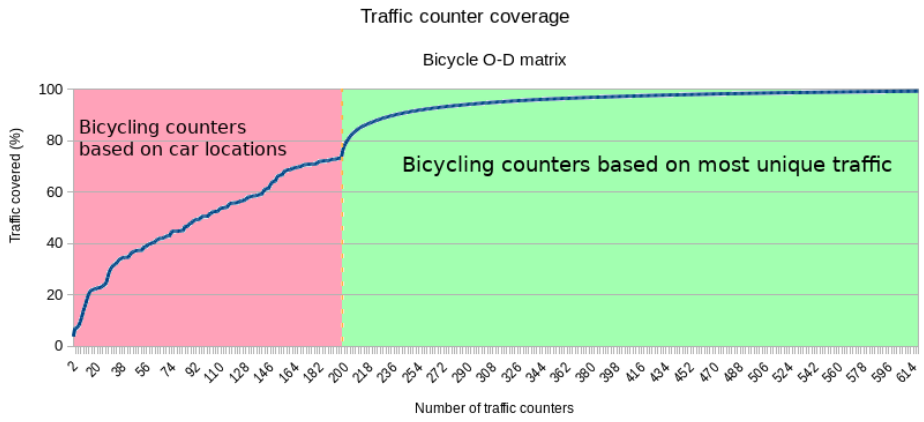


Figure 6.2: Chart of unique bicycle traffic covered. The first 200 points are based on the top 75 locations with the most unique car traffic. The difference in points is due to the fact that for proper coverage we include all links parallel to the car locations.

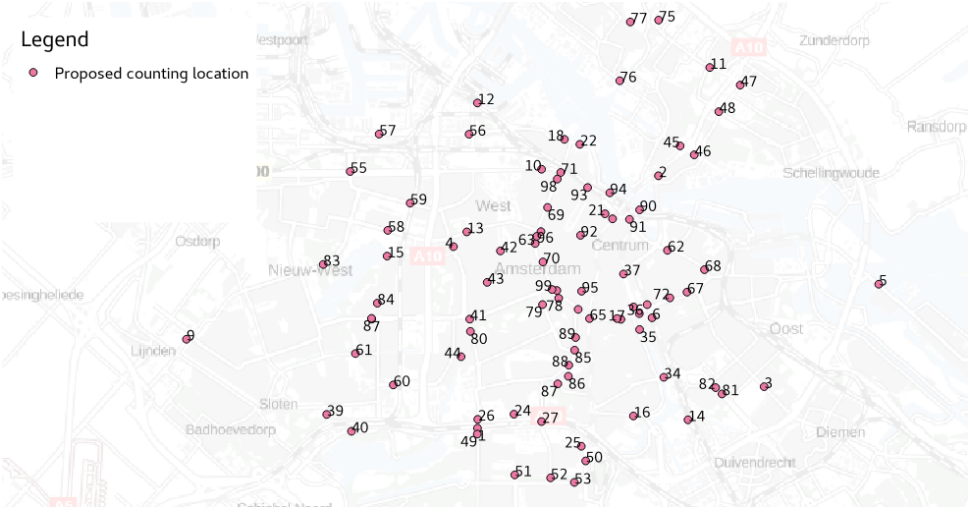


Figure 6.3: Map with visualization of the top 100 traffic counting locations capturing the highest number of unique traffic. Each digit indicates the priority, firstly we prioritize all locations that are permanently covered, then the locations within screen-lines and finally a free-selection. Each point within a group is ordered by the number of unique traffic it captures.

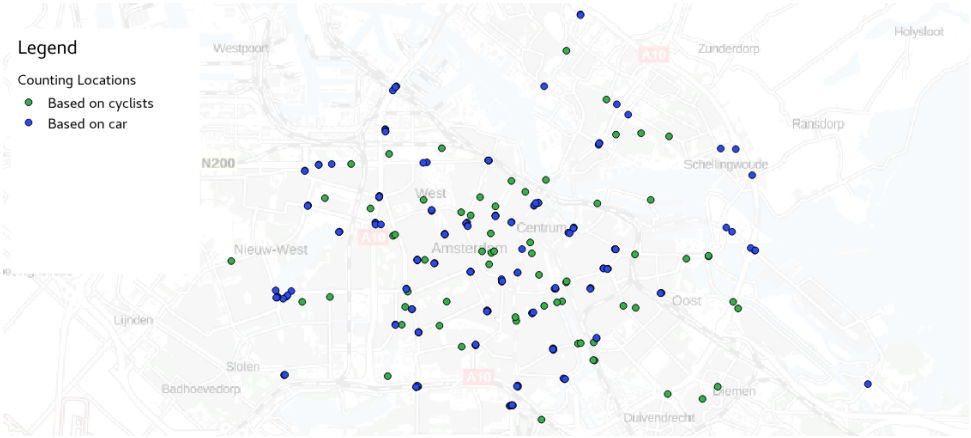


Figure 6.4: Map with the resulting counting locations for bicycles. The green points mark locations added for just bicycling; the blue points indicate that they were chosen based on car traffic patterns.

6.4.4. RESULTS

In the first step to obtain the best locations, we calculated the coverage based on 22 links in the network at locations such as tunnels that are permanently covered by loop counters; these 22 links are sorted based in descending order on the most remaining unique traffic captured. The next 77 edges were organized in 6 screen-lines. Originally the city of Amsterdam proposed 17 possible screen-lines, which were then sorted using the algorithm described; the results of this process are described in table 6.1. From this list the city selected the ten screen-lines covering the most traffic. The 24th to 54th counting point are selected within these screen-lines in a descending order of most unique traffic. The remaining counting points were selected using the unique traffic remaining.

The top 100 locations for cars were used in our calculations for bicyclists; this resulted in 197 "locations" since OpenStreetMap had many segments parallel to each selected link in the car network.

6.5. CONCLUSION

Our idea to assign traffic from O-D matrices to a network to determine locations that capture the most unique traffic in descending order worked well. The flexibility we have in selecting locations allowed us to respond quickly to additional requirements that came later on in this study. Requirements such as organization of counting locations within screen-lines was solved by determining how much traffic each screen-line captures with taking into effect of screen-lines close to each other. Since any point can be arbitrarily inserted into the process, allowed to take into account the preference for historical locations for trend-lines and the preference to combine cycle with car locations all came in the end phase of this project.

For future research it would be interesting to also find the optimal screen-lines, or a weighting function to take other factors into account besides capturing more traffic. In this study we used the screen-lines that were used in previous traffic studies since they

came with the added benefit of providing new data on historical trends, but it should be possible to find screen-lines that separate more traffic with fewer traffic counters, since we know for each segment where each vehicle is coming from and is heading to.

7

A Comparison of Approaches for the Time Series Forecasting of Motorway Traffic Flow Rate

Congestion forms a large problem in many major metropolitan regions around the world, leading to delays and societal costs. Congestion is generally associated with reduced average speed at a high traffic flow rate. This traffic flow rate is defined as the number of vehicles that pass a certain location at a given time. The modeling and prediction of this traffic flow rate may lead to valuable insights that may be used to reduce congestion and societal costs. This study aims to predict the traffic flow rate for 41 different locations in and around Amsterdam, The Netherlands. Using TBATS, SARIMAX and LSTM models, among others, the traffic flow rate of these locations has successfully been modeled. These models may provide accurate predictions for the future flow rate, which may be useful for the identification of infrastructure bottlenecks and the scheduling of maintenance. Considering the dramatic effects of the COVID-19 pandemic on the traffic flow rate, the inclusion of 2020 data with a number of external factors, could lead to an improvement of the results and the ability to model the future effects of the pandemic.

van der Bijl B., Gijsbertsen B., van Loon, S., Reurich Y., de Valk, T., Koch, T. and Dugundji E.R. (2022). A Comparison of approaches for the time series forecasting of motorway traffic flow rate. Accepted for The 13th International Conference on Ambient Systems, Networks and Technologies.

7.1. INTRODUCTION

Congestion is known to have increased substantially in major cities over the course of the past few decades. Therefore, it is essential that governments stay aware of any trends regarding the traffic flow rate in the major cities. This awareness enables governments to foresee any bottlenecks in the infrastructure in the long term. These bottlenecks lead to a lot of congestion, which results in immense costs. The Dutch institute for mobility policy (KiM) estimated the societal cost of congestion on the primary highway network at 3.3 to 4.3 billion euros in 2018 Kennisinstituut voor Mobiliteit, 2019. Accurately predicting and preventing the peaks in the traffic flow rate could lead to a huge cut in these costs.

Based on mobility data in 2015, the Dutch knowledge platform on traffic and mobility, CROW Voerknecht, 2016, forecasts that the increase in traffic flow rate will be especially problematic in the city of Amsterdam with its lack of space and large number of narrow streets. Therefore, this study is focused on the analysis of the current traffic flow rate in Amsterdam as well as the prediction of this traffic flow rate for the future. These predictions are done by identifying and using patterns in the historical data.

The prediction of the future traffic flow rate can be used for various purposes. The short-term predictions, for instance, can be used to find a route through Amsterdam that avoids congestion. As previously mentioned, the long-term predictions can be used by the government or the municipality of Amsterdam in the identification of bottlenecks to take appropriate actions. These long-term predictions can also be used to schedule maintenance without drastically affecting the traffic flow rate. The analyses and predictions can also be used to investigate the traffic flow rate with respect to the COVID-19 pandemic.

Section 7.2 contains a description of the related literature, about papers that describe similar or relevant studies. The section that follows, Section 7.3 provides a description of the case study. In this section, the given data set and the practical relevance of the modeling of the traffic flow rate are described. Section **??** describes a review of the literature. This review is focused on different types of time series models, the evaluation of these models and possible extensions of the given data set. Section 7.4, contains an overview of the results. These results are discussed in Section 7.5. Lastly, concluding words are given in Section 7.6 and future work is proposed in Section 7.7.

7.2. BACKGROUND

With regard to related literature, multiple papers have addressed the issue of increasing traffic flow rate in cities. For instance, short-term traffic speed predictions have been made using univariate and multivariate neural network models as well as ARIMA models Vlahogianni and Karlaftis, 2013. With the inclusion of daily rainfall for the road in question, Vlahogianni and Karlaftis (2013) showed that neural network models allowed for more accurate time series predictions than classical time series forecasting methods. Rahman (2020) investigated k-nearest neighbors, support vector machine and neural network models in the prediction of traffic flow. This study accentuated the value of weather data in the prediction traffic flow for a one-month period of time with an interval. In addition, Zhou et al. (2017) investigated short-term predictions for traffic flow using multiple boosted algorithms such as an AdaBoost stacked autoencoder. In an-

other study, Xie et al. (2010) used SVM and ARIMA models to evaluate the performance of Gaussian process models in the prediction of traffic volume for four locations around Seattle, USA. This study shows that SVM and Gaussian process models generally outperform ARIMA models. T. Ma et al. (2020) investigated the modeling of neural network residuals with ARIMA models. Ensembling these models led to a notable performance increase of 8.9–13.4% in the prediction of the traffic flow for 44 locations in Tel Aviv, Israel, with at least six months of data per location. In addition to SARIMA models being suitable for the prediction of the traffic flow, Shekhar and Williams (2007) suggest that filtering techniques such as the Kalman, RLS and LMS filters may boost predictive performance, achieving similar performance to machine learning models. Moreover, Shekhar and Williams (2007) establishes that SARIMA models are fairly insensitive to parameter tuning and, therefore, robust.

In contrast to a large segment of the available literature, this chapter aims to investigate the application of a broad spectrum of time series and machine learning models in the short and long-term prediction of the traffic flow rate for a number of separate locations in Amsterdam, The Netherlands. These predictions disregard spatial information about the data and are used to highlight the effects of the COVID-19 pandemic as well as potential future bottlenecks in and around Amsterdam.

Considering that the traffic flow rate depends on time, it can be interpreted as a time series. There are multiple models that can be used for time series prediction. A number of these models are described in this section. After the elaboration of the different models, the evaluation techniques of time series models are discussed. These evaluation techniques are used to judge the quality of the fit of the model, as well as to compare the performance of the different models. Furthermore, the input data may be enriched with exogenous factors. This section also elaborates on which factors are suitable for this.

7.2.1. MODELS

SEASONAL NAIVE MODEL

Seasonal naive models are suitable for data with clear seasonal components R. J. Hyndman and Athanasopoulos, 2018. For this model, the predicted value is equal to the last observed value from the same moment of the previous season. Considering that the input data show multiple seasonal components, as described in Section 7.3.2, this approach may be limited. Another limitation of this model is that potential trends are not used in the prediction. Disregarding the limitations, the simplicity of the model and the seasonal patterns of the input data, the seasonal naive model will be used as a baseline throughout this research.

AUTOREGRESSIVE INTEGRATED MOVING AVERAGE MODELS

Autoregressive integrated moving average, or ARIMA, models are a combination of autoregressive (AR) and moving average (MA) models. The order p in the AR model represents the number of historical values that are used in a linear combination to form the next value. The order q in MA models represents the number of historical values that are used in the moving average. The orders p and q can be determined using the partial autocorrelation function (PACF) and the autocorrelation function (ACF) plots, respectively. The next value X_t can be defined using the following equation:

$$X_t = \sum_{j=0}^{q} \beta_j Z_{t-j} + \sum_{i=1}^{p} \alpha_i X_{t-i} + Z_t \tag{7.1}$$

In this equation, $\{Z_t\}$ is a white noise time series. $\alpha_1, \alpha_2, ..., \alpha_p$, and $\beta_0, \beta_1, ..., \beta_q$ are the weights that describe the contribution of the previous p and q values, respectively.

ARIMA models are most suitable for the modeling of time series that contain trends. The modeling of time series with seasonal patterns can be overcome by differencing. This differencing is described by parameter d, that ARIMA adds to the AR and MA models, resulting in an order (p, d, q) Esprabens and Arango, 2020.

ARIMA models require different modeling assumptions to be made. In the first place, the time series is required to be stationary. For stationary time series, $E(X_t)$ and $E(X_t X_{t+h})$ exist and do not depend on t. This means that the time series is consistent over time, indicating that the time series can not contain trends or seasonal patterns. An Augmented Dickey-Fuller test can be used to test whether a given time series is stationary. In second place, the residuals are required to be normally distributed. To determine whether the residuals are normally distributed, a comparison can be made between a kernel density estimation of the distribution of the residuals and the normal distribution. Further, a correlogram of the residuals can be consulted to determine whether there are still patterns present in the residuals. If this is not the case, it can be concluded that the ARIMA model is an appropriate fit.

SEASONAL AUTOREGRESSIVE INTEGRATED MOVING AVERAGE MODELS

ARIMA models are used to make a non-stationary time series stationary by adjusting the trend. Seasonal autoregressive integrated moving average, or SARIMA, models are an extension of this, in which a seasonal component is also considered. This means that a SARIMA model is used to make a non-stationary time series stationary by removing the trend and seasonality Esprabens and Arango, 2020. This results in the addition of seasonal parameters P, D, Q and s, where s is the number of observations in a season. With that, a SARIMA model can be described by the order (p, d, q)x$(P, D, Q)[s]$. Similarly to the ARIMA model, the ACF and PACF plots can also be used to determine the values for P and Q.

A SARIMA model uses differencing at a lag equal to the number of seasons (parameter s) to remove additive seasonal effects Cowpertwait and Metcalfe, 2009. The best way to find the most suitable values for the parameters, is by trying a large number of models with different parameter values Brownlee, 2019. These parameter values can be determined using a grid search.

Additionally, SARIMA can also be implemented with exogenous features, resulting in a SARIMAX model Khandelwal, 2020. These exogenous features, as the name states, have a different origin than the original time series. This means that the features describe other characteristics of the observations than the time series does. The addition of these features may lead to a more precise forecast of the time series.

(T)BATS MODELS

(T)BATS is an acronym to describe the key features of the model: (**T**rigonometric seasonality), **B**ox-Cox transformations, **A**RIMA errors, **T**rends, **S**easonal components. The

way in which these model components are used is fully autonomous NHS England and NHS Improvement, 2020.

TBATS models use exponential smoothing to model highly seasonal data. Whereas most popular models can incorporate at most one seasonal pattern, a TBATS model is known to incorporate multiple, complex seasonal patterns. TBATS models can incorporate seasonal patterns of which the frequency is not constant Pentaho, 2020. According to Skorupa (2019), each seasonality in a TBATS model is modeled by a trigonometric representation based on a Fourier series. A Fourier series is an expansion of a periodic function in terms of an infinite sum of sines and cosines. It is predominantly used to break up periodic functions to calculate them individually Weisstein, 2020. The Box-Cox transformations that are incorporated in the model, enable the model to handle non-linearity in the data. This stabilizes the variance in the predictions Chou, 2017.

One major drawback of TBATS models is that forecasting may be slow, especially for long time series R. J. Hyndman and Athanasopoulos, 2018. Another drawback of TBATS models is that other explanatory variables may not be added, potentially limiting the overall performance. The long-term predictions of TBATS models are thereby not always adequate. Furthermore, TBATS models are used under the assumption that the residuals are normally and independently distributed Chou, 2017. According to R. J. Hyndman and Athanasopoulos (2018), the autonomous nature of the model components may thus lead to predictions that are not useful.

RECURRENT NEURAL NETWORKS

A recurrent neural network, or RNN, is a neural network that is specifically designed for the analysis of sequences. A specific subsection of RNN's that are proven to be useful in time series modeling are Long Short-Term Memory models, or LSTM models. These models have a similar structure to neural networks, where there are input and output layers, with at least one hidden layer to connect them both. The difference between neural networks and LSTM models is the option to make connections between neighboring nodes within the same (hidden) layer. The connections make it possible to attain and retain 'memory'.

When training LSTM models, parameter-tuning is crucial. This means that optimal values need to be found for the parameters of the LSTM. In the first place, the number of input nodes determines how many historical values are used. In second place, the number of output nodes determine the number of future values that are predicted. When more output nodes are present, the accuracy and reliability of the prediction will most likely decrease. The number of hidden layers determines the number of node layers between the input and output layers. Each hidden layer contains a given number of (hidden) nodes, each with a non-linear activation function. Increasing the number of layers and nodes leads to an increase in complexity. This means that a trade-off has to be made between the complexity and the training time. The training of a LSTM model is done for a number of epochs, which is equal to the number of times that the model will iterate through the entire data set. The most suitable values for these parameters can be found using a grid search, where the different models are evaluated using the evaluation metrics that are discussed later.

LSTM models generally have one output node, giving one predicted value. In time series prediction, however, it is essential to predict further into the future. For that reason,

the rolling update method is used. This method is also known as walk-forward validation Tutorialspoint, 2021. The essence of the method is to use observed data as well as predictions to predict future values. This prevents leakage of the validation data and it enables the LSTM model to predict farther into the future.

ALTERNATIVE MODELS

Other models that have been considered for time series prediction are exponential smoothing, the Holt-Winters method, autoregressive moving average models and variations thereof, vector autoregressive moving average models and variations thereof, k-nearest neighbors, convolutional neural networks and graph neural networks. These models have not been discussed as they are too complex, lack flexibility or the ability to accommodate complex seasonal patterns.

7.2.2. EVALUATION OF MODELS

The evaluation of a model is crucial for the understanding of the performance of the model and the comparison of different models. There is a large array of metrics that can be used for this task. These metrics can be categorized as follows:

- Metrics to indicate the distance between the predicted and actual values
- Metrics that show the fraction of the variance in the data that is explained by model
- Metrics that assist in the trade-off between complexity and likelihood

Some of these metrics are more suitable than others for the evaluation of time series forecasting. In this case, the MAE, or mean absolute error, is the best metric to use for the comparison of different models JJ, 2016. The MAE is a measure of the distance between the predicted and actual values, calculated using the following equation:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n} \tag{7.2}$$

Moreover, the AIC is used to select the most appropriate orders for the ARIMA and SARIMA models. This measure penalizes the use of a large number of parameters and reduces the probability of overfitting.

7.3. CASE STUDY

7.3.1. DESCRIPTION OF DATA

The data set that has been provided by the Dutch data portal for road traffic (NDW) Nationaal Dataportaal Wegverkeer, 2021 and consists of 9,619,639 observations with 22 columns. These observations describe the flow rate of traffic, in the number of vehicles that pass over a road sensor per hour, from 41 locations around Amsterdam. These locations are shown in Figure 7.1. It can be seen that the locations describe major motorways around Amsterdam as well as a few smaller roads in the city center. The data set provides hourly data for the 41 locations for five years from the 1st of January 2016 up until the 1st of January 2021. The data describes an hourly aggregation of the number of vehicles, the average speed and an optional categorization by the length of those vehicles.

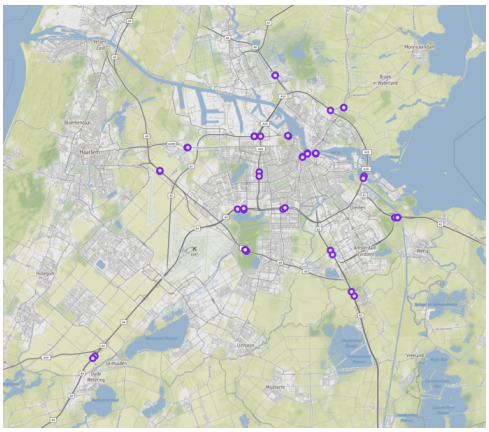


Figure 7.1: Visualization of the sensor locations showing a wide variety of different sensor locations in and around Amsterdam.

7.3.2. DATA ANALYSIS

To determine which models are applicable to the provided data, an extensive data analysis is carried out. The data analysis will also provide other valuable insights regarding the traffic flow rate in Amsterdam.

Since there is a large number of locations, it is not viable to show the analyses for all locations. Therefore, two locations have been chosen. The two locations that have been chosen are the A10 motorway at `RWS01_MONIBAS_0101hrr0189ra` and the A4 motorway at `RWS01_MONIBAS_0041hrr0197ra`. These locations show relatively large traffic intensities and a large variance in traffic speed. It is assumed that large motorways, such as these, have a larger effect on the traffic flow rate of Amsterdam than smaller roads do. These two locations show a clear relationship of the average speed and traffic flow rate. Most other locations show similar effects, which is why we expect that any conclusions will generalize well. The A10 and A4 locations that have been chosen for the data analysis, will also be used later in the project for the validation of the different models. This drastically simplifies the validation of models.

One metric that gives insight into the traffic flow rate for certain locations, is the density of vehicles. The density of vehicles is defined by the total number of vehicles, divided by the average speed and the total number of lanes, resulting in the number of vehicles per kilometer. Figure 7.2 shows the average vehicle density versus the average speed for the A10 location. It can be seen that a low speed yields a higher density of vehicles. This means that vehicles are closer together at lower speeds, suggesting that there is congestion. The A4 location showed similar patterns regarding the vehicle density.

The average traffic flow rate can also be used to estimate which locations have the potential to be future bottlenecks for traffic in Amsterdam. These locations can be identified by the fraction of time in which they are at their maximum capacity. Unfortunately, the locations that are described by the data set have extremely diverse characteristics and are, therefore, remarkably difficult to compare. This is especially difficult when it comes to speed and flow rate. For that reason, only the roads and motorways with a

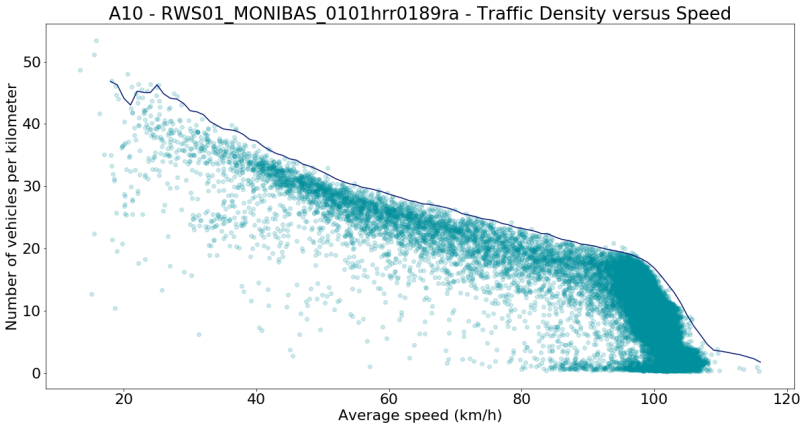


Figure 7.2: Scatter plot of the average speed versus traffic density (number of vehicles per kilometer) on the A10. It can be seen that a low speed yields a higher density of vehicles, indicating that vehicles are closer together at lower speeds and suggests that there is congestion.

maximum speed of 100 km/h, have been analyzed regarding future bottlenecks. In Figure 7.2, it can be seen that the largest densities are reached for speeds of less than 30 km/h. With that, potential future bottlenecks can be identified by investigating which motorways spend the largest fraction of time at a speed of less than 30 km/h.

Figure 7.3 shows the average traffic flow rate per year for the A10, with the standard deviation plotted around the average. Since the different years approximately follow the same patterns, it can be concluded that there is a yearly seasonal component. It can also be seen there are no clear upwards or downwards trends that need to be considered. Furthermore, it can be seen that 2020 significantly deviates from the other years. This is a result of the COVID-19 pandemic. Especially the first national lockdown from March of 2020, shows a remarkable decrease in the traffic flow rate. This deviation will also have a notable impact on the modeling of the data and the forecasting of the future traffic flow rate. For that reason, it has been decided that model validation will be carried out in 2019 rather than 2020. These patterns were also visible for the A4 location.

The plot in Figure 7.4 shows the progression of the average traffic flow rate per hour of the week for the A10. Here, 2020 is disregarded as the COVID-19 pandemic affects the clarity of seasonal patterns. The plot shows a clear daily seasonal component. It can be seen that the traffic flow rate is highest in the morning and in the afternoon. This is shown by the two peaks in the graph of each day. Furthermore, it can be seen that the traffic flow rate is by far the lowest during the nights. The plot also clearly shows that the traffic flow rate is far lower on Saturdays and Sundays than it is the rest of the week. It can also be observed that the nights during the weekend are slightly less quiet than during the rest of the week. Another pattern that can be identified is that the morning rush hour contains a higher peak in traffic flow rate than the evening rush hour does. The traffic flow rate is slightly more spread out during the evening rush hour. These weekly and

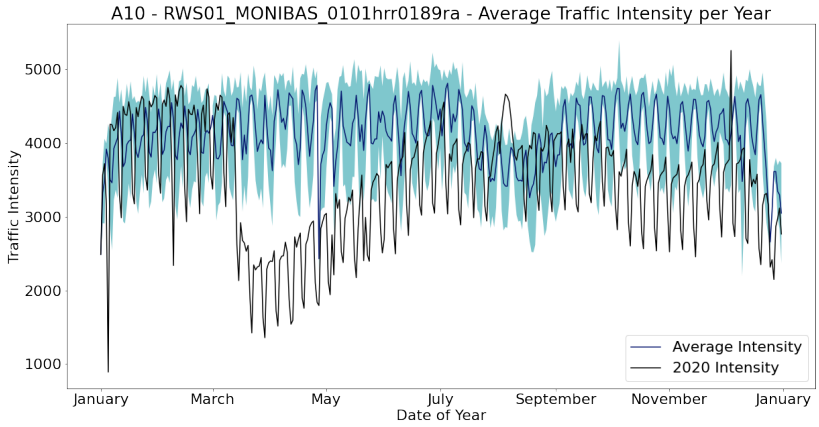


Figure 7.3: Plot of the average flow rate per year of the A10. Multiple seasonal patterns can be observed and it can be seen that the traffic flow rate for 2020 deviates from the general patterns. This is a result of the measures to reduce mobility in order to stop the spread of COVID-19.

daily patterns were also visible for the A4.

It is expected that the different seasonal components that have been identified, may interfere with each other. This may result in difficulties when modeling the data.

7.3.3. EXTENSION OF DATA

Next to the data that have been provided, there may be more characteristics of the different locations that also affect the traffic flow rate.

One of the external features that may affect traffic flow is the weather US Department of Transportation, 2018. Weather characteristics such as precipitation, fog and wind affect the traffic flow negatively. For that reason, obtaining weather data for the different locations in Amsterdam, may prove to be extremely valuable.

Next to the weather, large events in Amsterdam may also contribute to the traffic flow rate. Therefore, information regarding events in venues such as the RAI Convention Center, the Johan Cruijf Arena or the Ziggo Dome, may provide additional value to the traffic flow rate data.

Considering that the COVID-19 pandemic has had a remarkable effect on the traffic flow rate for 2020, using COVID-19 statistics to enrich statistics may provide more precise predictions of the traffic flow rate. The following statistics, among others, may provide valuable information in the prediction of the traffic flow rate: the number of positive COVID-19 tests, the number of COVID-19 patients admitted to the hospital, the number of COVID-19 patients admitted to intensive care units, the number of deaths per day as a result of COVID-19 infections. Since the Dutch government bases its COVID-19 measures on these statistics, it is assumed that these statistics will affect the traffic flow rate.

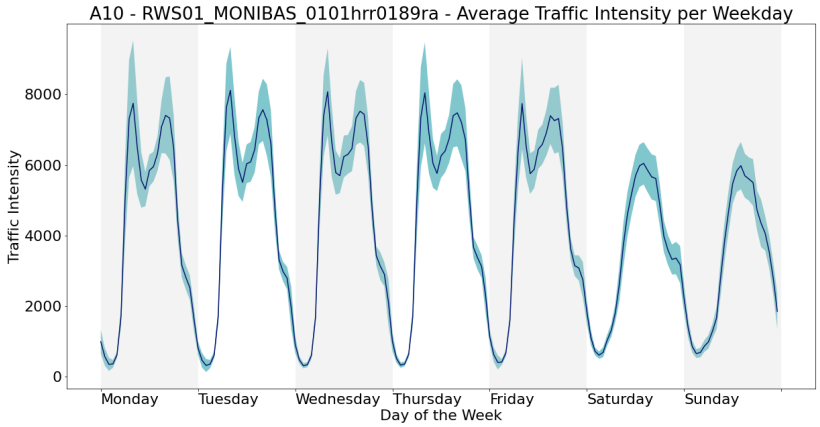


Figure 7.4: Plot of the average traffic flow rate per day of the week of the A10. Clear daily and weekly patterns can be observed, with peaks during the morning and evening rush hours.

7.4. RESULTS

In this section, the results of the modeling stage are thoroughly discussed. As previously mentioned, the models were initially fitted to hourly data. This worked adequately for the simple models, but the computational limits formed a problem when training larger, more complex, models. Therefore, the complex models were only applied to the daily-aggregated data.

As mentioned in Section 7.3.2, the seasonal naive method forms a baseline for the performance of the models. Therefore, the models that underperformed with respect to the seasonal naive method, have been disregarded.

7.4.1. HOURLY RESULTS

SEASONAL NAIVE MODEL

The seasonal naive model achieves a MAE of 2999.63 for the A10 and 1898.77 for the A4. Since the seasonal naive model is used as a baseline model, these MAEs represent the baseline statistics for the validations for the traffic flow rate in 2019.

TBATS MODEL

The TBATS model achieves a MAE of 777.68 for the A10 and 578.10 for the A4. It is expected that the performance of TBATS models is better for short-term predictions, than for long-term predictions. To investigate this hypothesis, the MAE for the first six months of the validation was compared to the last six months of the validation. These MAEs were extremely similar, which is why the expectation was incorrect.

The TBATS models were trained with seasonalities of 24, 168 and 8766 hours. These seasonalities are equal to a day, 7 days and 365.25 days (a year with the consideration of leap years). During the training of the model, Box-Cox transformations and ARIMA

errors were used. The data were also transformed using a log-transformation before being entered into the model. These measures ensure that no negative predictions can be made.

To determine whether the TBATS fit is adequate, the modeling assumptions need to be verified. In the case of TBATS models, the residuals are required to be normally distributed. According to kernel density estimation plots, the residuals are approximately normally distributed for the A10 and A4 locations. Therefore, it can be stated that the TBATS models fit the data adequately.

7.4.2. DAILY-AGGREGATED RESULTS

SEASONAL NAIVE MODEL

The seasonal naive model achieves a MAE of 11110.53 for the A10 and 6880.92 for the A4 on daily-aggregated data. These values form the baseline for the daily-aggregated models.

(SEASONAL) ARIMA MODEL

ARIMA models were not appropriate for the hourly traffic flow rate due to the interference of different seasonal patterns. Since the daily-aggregated data have less seasonalities, the ARIMA model may be suitable for the daily-aggregated data. In Section 7.3.2, it was concluded that the traffic flow rate contains both a weekly pattern and a yearly pattern. Therefore, a difference of order 365 and a difference of order 7 were used on the log-transformed time series of the A10 location. These differences remove any seasonal patterns and result in a stationary time series. This stationarity is tested using the Augmented Dickey-Fuller test, resulting in a p-value of $1.21 \cdot 10^{-14}$. This means that with an α-value of 0.05, the null-hypothesis of the time series being stationary, is rejected. Therefore, ARIMA models may not be suitable for the modeling of the data.

Similarly to the A10, the differencing transformations were also carried out for the A4. For the resulting time series, the Augmented Dickey-Fuller test led to a p-value of $1.23 \cdot 10^{-17}$, indicating that the null-hypothesis is rejected once again. This implies that the data are not stationary. Just as for the A10 that was described previously, a standard ARIMA model may not suffice for the modeling of the traffic flow rate at the A4 location.

The PACF of the transformed time series of the A10 shows that there is a periodic deviation around lag seven. This indicates that there may still be some weekly seasonality in the transformed data. This seasonality explains why the transformed time series is not stationary. A seasonal ARIMA (SARIMA) model could potentially solve this issue. This leads to a SARIMA model of the order $(2,0,1)$x$(1,0,1)[7]$. The same conclusions hold for the A4, leading to a SARIMA model of the order $(1,0,1)$x$(1,0,1)[7]$.

The MAE for the SARIMA model for the A10 is equal to 7976.32. The MAE for the A4 is equal to 5118.73, which is a notable improvement with respect to the baseline model.

As mentioned in Section 7.2.1, SARIMA models require certain modeling assumptions. With that, the residuals need to be normally distributed and uncorrelated. These assumptions are verified using a plot of the fitted values against the observed values, a kernel density plot of the residuals and a correlogram. These plots suggest that the residuals are indeed normally distributed and uncorrelated. This holds for the A10 and

A4 locations. Therefore, it can be concluded that the SARIMA models fit the data adequately.

SARIMAX MODEL

Next to the SARIMA model that has been elaborated in the previous section, a SARIMAX model has been developed. This model is equal to the SARIMA in the way that it was developed, with one crucial difference: namely, the SARIMAX model incorporates weather data from Amsterdam. As mentioned in Section 7.2.1, the weather seems to affect the traffic flow rate.

To determine the influence of the weather on the traffic flow rate, the correlation between the traffic flow rate and several weather characteristics has been investigated. This analysis showed little correlation between the traffic flow rate and the following daily weather characteristics: wind gusts, average temperature in degrees Celsius, number of sun hours, daily fraction of fog, daily fraction of rain, amount of snow and ice formation.

The SARIMAX model that has been fitted to the A10 has an order of $(2,0,1)$x$(2,0,1)[7]$. This model results in a MAE of 11110.43 for the 2019 validation of the model. For the A4 location, the SARIMAX has an order of $(1,0,0)$x$(1,0,1)[7]$. This model results in a MAE of 6880.82 for the 2019 validation of the model. Considering the MAE of the SARIMAX models, it can be stated that the SARIMA model outperforms the SARIMAX model that incorporates the weather data. This indicates that the chosen weather variables do not have the desired effect on the traffic flow rate.

Furthermore, the use of weather data is fairly limited as future forecasts of Amsterdam weather are not provided. This means that only past weather data can be used to predict the future. Past weather data are expected to be invaluable with regard to future predictions.

TBATS MODEL

The TBATS model achieves a MAE of 5365.72 for the A10 and 16380.73 for the A4. Considering the results of the baseline model, it can be stated that the TBATS model is adequate for the A10 location. However, it can also be seen that the TBATS model is far from adequate for the A4 location.

The TBATS models were trained with seasonalities of 7 and 365.25 days. These seasonalities are equal to a week and a year with the consideration of leap years. During the training of the model, Box-Cox transformations and ARIMA errors were used. Log-transformations of the data did not further improve the performance.

TBATS models require the residuals to be normally distributed. Using kernel density estimation plots, this assumption has been verified for both the A10 and A4 locations. Therefore, the TBATS models can be considered to be an adequate fit.

The results from the TBATS model were inconsistent based on the MAE for the A10 and A4 locations. The MAEs show that the TBATS model is not always better than the baseline seasonal naive model. Therefore, a model is trained for a number of other locations. The MAEs for these locations are used in the final decision regarding the use of TBATS models for the modeling of the traffic flow rate. This decision led to the use of TBATS models to model the traffic flow rate.

LSTM MODEL

The final type of model that is applied to the daily-aggregated data is the LSTM model. As mentioned in Section 7.2.1, a grid search could be used to find the most suitable parameters for the LSTM models.

The parameters that were investigated using the grid search are described in Table 7.1. This grid search was conducted for both the A10 and A4 locations. The shape of the input layer of the LSTM was equal to (365,1), meaning that the LSTM will receive 365 input values. This indicates that the preceding 365 days of data are used to predict the next day.

Table 7.1: LSTM Grid Search Parameters

Parameter	Possible Values
Number of Epochs	10, 50, 100, 150
Batch Size	1, 5, 10, 20, 50
Number of Nodes Layer 1	1, 3, 5, 10, 20, 50, 100.
Number of Nodes Layer 2	0, 1, 3, 5, 10, 20, 50, 100
Activation Functions	hyperbolic tangent, sigmoid, ReLu

The grid search led to different models for the A10 and A4 locations. In the trade-off between complexity and performance, however, a simpler model was selected. This simpler model performed nearly as well as the model that was selected using the grid search. Moreover, the simpler model was identical for both the A10 and A4 locations. This indicates that a model with the suggested architecture generalizes well and is likely suitable for other locations. The characteristics of the final model and training procedure can be described as follows:

- Number of epochs: 100
- Batch size: 1
- Number of nodes layer 1: 20
- Number of nodes layer 2: 0 (indicating that the model will only have one hidden layer)
- Activation functions: hyperbolic tangent

Using the rolling update method, or the walk forward validation that was described in Section 7.2.1, the two models were validated. This validation led to a MAE of 4503.00 for the A10 and 3423.48 for the A4.

7.5. DISCUSSION

Considering the results mentioned in Section 7.4, the MAE on the hourly data of 2019 is shown for a variety of models in Table 7.2. In this table, a distinction has been made between the A10 and A4 locations.

Table 7.2: Hourly Results

Model	Location A10 MAE 2019	Location A4 MAE 2019
Seasonal Naive	2999.63	1898.77
TBATS	**777.68**	**578.10**

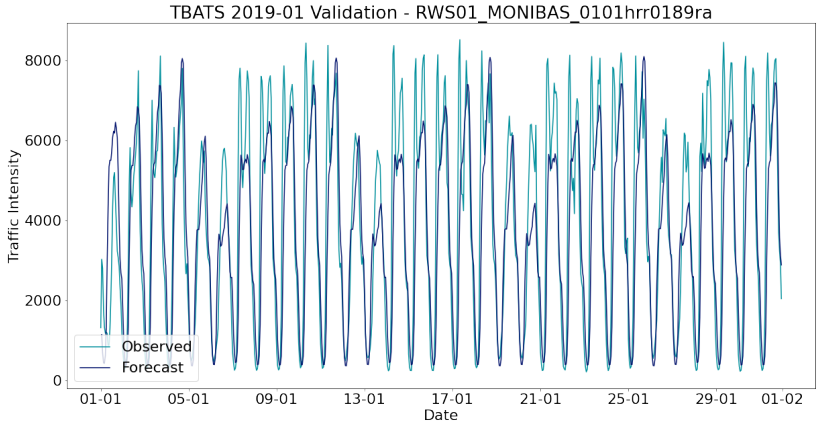


Figure 7.5: The validation in 2019 of the A10 motorway using the TBATS model with hourly data. The forecast seems to follow the observed data fairly well, with the exception of the underestimation of the peaks. The ascending nature of the peaks is not found in the observed data and is likely a result of the sinusoidal nature of the TBATS model. This model performed 74.1% better than the seasonal naive baseline model, regarding the MAE.

The best model that has been fitted to hourly data is the TBATS model. On average, this model performed 72% better than the seasonal naive baseline model, regarding the MAE.

In Figure 7.5 and 7.6, the 2019 validation of the TBATS models has been plotted. It can be seen that the forecasts follow the observed data fairly well. An exception to this is the fact that the forecast nearly always underestimates the peaks of the observed data. One possible cause for this is that the traffic flow rate of 2018 was slightly lower than that of 2019. This trend was not as present in preceding years, which is why the TBATS model may not incorporate it sufficiently. As a result, the MAE is still notably high.

In Figure 7.5 and 7.6, it can also be seen that the forecasts contain an unusual pattern, where the traffic flow rate is predicted to be low at the start of a week and higher towards the end. When this pattern is compared to the observed traffic flow rate, it can be seen that this pattern is not equal to the observed weekly patterns. This is likely due to the sinusoidal nature of the TBATS model.

Due to the limit of computational resources, the implementation of other models

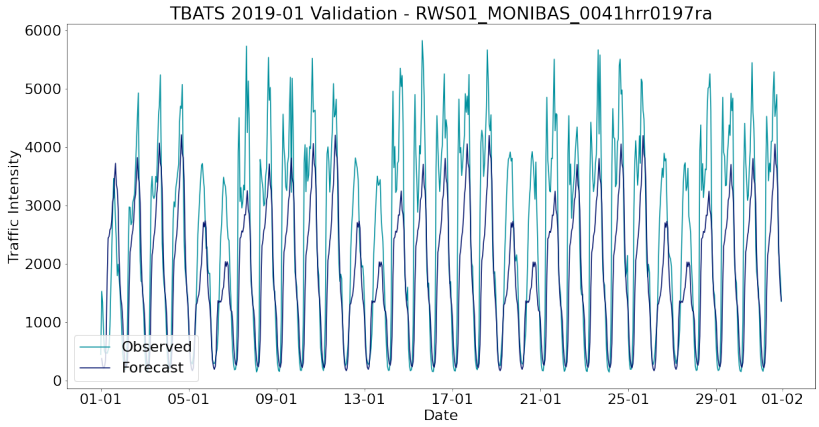


Figure 7.6: The validation in 2019 of the A4 motorway using the TBATS model with hourly data. The predicted values seem to follow the observed data fairly well, with the exception of the underestimation of the peaks. The ascending nature of the peaks from Monday until Friday is not found in the observed data and is likely a result of the sinusoidal nature of the TBATS model. This model performed 69.6% better than the seasonal naive baseline model, regarding the MAE.

to the hourly data was not possible. Therefore, a number of models have been implemented on daily-aggregated data. The MAE on the daily-aggregated data of 2019 is shown for a variety of models in Table 7.3. The Holt-Winters method is omitted from this table as it underperformed with respect to the seasonal naive model.

The best model that has been trained on the daily-aggregated data is the LSTM model. On average, this model performed approximately 55% better than the seasonal naive baseline model, regarding the MAE. Figure 7.7 shows the daily validation of 2019 for the A10 using the LSTM model and Figure 7.8 shows this validation for the A4. It can be seen that the two LSTM models describe the data in both figures fairly well.

Table 7.3: Daily-Aggregated Results

Model	Location A10 MAE 2019	Location A4 MAE 2019
Seasonal Naive	11110.53	6880.92
SARIMA	7976.32	5118.73
SARIMAX	11110.43	6880.82
TBATS	5365.72	16380.73
LSTM	**4503.00**	**3423.48**

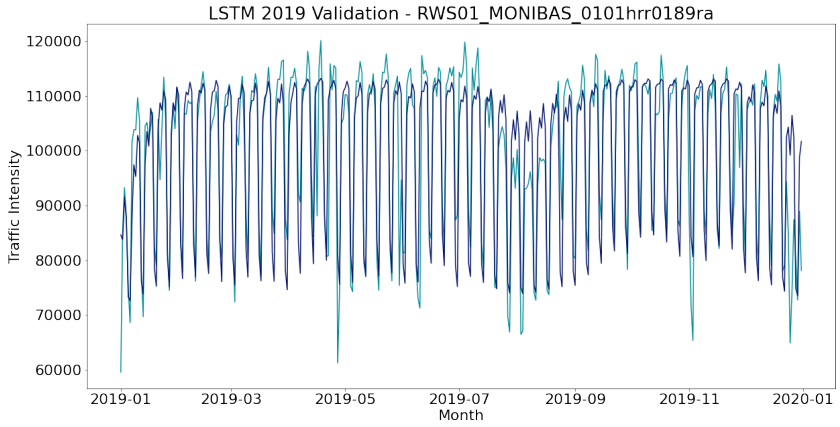


Figure 7.7: The validation in 2019 of the A10 location using the LSTM model with daily-aggregated data. The forecast seems to follow the weekly and yearly seasonal patterns fairly well. This model performed 59.5% better than the seasonal naive baseline model, regarding the MAE.

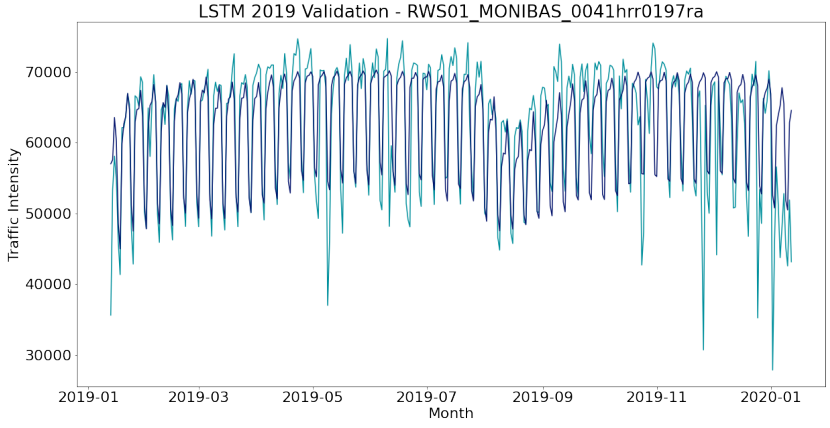


Figure 7.8: The validation in 2019 of the A4 location using the LSTM model with daily-aggregated data. The forecast seems to follow the weekly and yearly seasonal patterns fairly well. This model performed 50.2% better than the seasonal naive baseline model, regarding the MAE.

7.6. CONCLUSION

Considering the results that were discussed in Section 7.5, it can be stated the modeling of the traffic flow rate has been carried out successfully. That said, the TBATS model is considered to be the most suitable model for the hourly traffic flow rate data. This model performs 72% better than the baseline model, the seasonal naive model. With respect to

the daily-aggregated data, the LSTM model is the most suitable. This model performs 55% better than the seasonal naive model on the daily-aggregated data.

7.7. FUTURE STUDY

There are numerous, interesting additions that could enrich the current research.

Firstly, the manner in which models are implemented could be improved. The validation of the models was done in 2019 for two locations. With this, it was assumed that the model performance would be similar for other locations. In practice, this turned out not to be the case in all instances. Therefore, it is suggested that the models should be validated on all locations in the future. This ensures that the chosen model performs well on all locations. Another way in which the implementation of models can be improved is through the use of multiple evaluation metrics. In this study, the mean absolute error was used. To complement this, the use of metrics such as root mean squared error (RMSE), mean absolute percentage error (MAPE) or adjusted R-square are recommended.

Secondly, there are a number of models that have not been implemented. As mentioned in Section 7.2.1, there are other models that are potentially useful in time series prediction. These models include, but are not limited to, k-nearest neighbors, convolutional neural networks and graph neural networks. In future research, the implementation of these models may lead to positive results in the prediction of the traffic flow rate. Lastly, more extensions of the data could be considered. Some possible extensions were mentioned in Section 7.3.3. In this research, weather data were used to enrich the daily-aggregated data for a SARIMAX model. Since the weather is subject to change during a day, it is expected that the use of hourly weather data may be beneficial for the predictions. Next to the weather data, data regarding large events as well as COVID-19 statistics could be used to further enrich the data. This enables the predictions to include the effects of COVID-19. The addition of live data streams in combination with dynamic systems, would also allow for more accurate short-term predictions, accounting for unforeseen circumstances.

These additions may lead to better model performance or improved insights regarding the traffic flow rate in and around Amsterdam.

8

PATH COMPLEXITY AND BICYCLE ROUTE CHOICE SET QUALITY ASSESSMENT

Everyday route choices made by bicyclists are known to be more difficult to explain than vehicle routes, yet prediction of these choices is essential for guiding infrastructural investment in safe cycling. Building route choice sets is a difficult task. Even including detailed attributes such as the number of left turns, the number of speed bumps, distance and other route choice properties we still see that choice set quality measures suggest poor replication of observed paths.
In this chapter we study how the concept of route complexity *can help generate and analyze plausible choice sets in the demand modeling process. The complexity of a given path in a graph is the minimum number of shortest paths that is required to specify that path.* Complexity *is a path attribute which could potentially be considered to be important for route choice in a similar way. The complexity was determined for a large set of observed routes and for routes in the generated choice sets for the corresponding origin-destination pairs. The respective distributions are shown to be significantly different so that the choice sets do not reflect the traveler preferences, this is in line with classical choice set quality indicators. Secondly, we investigate often used choice set quality methods and formulate measures that are less sensitive to small differences between routes that can be argued to be insignificant or irrelevant. Such difference may be partially due to inaccuracy in map-matching observations to dense urban road networks.*

Koch, T., Knapen, L., & Dugundji, E. (2019). Path complexity and bicyclist route choice set quality assessment. Personal and Ubiquitous Computing, 25.1 (2021): 63-75.

8.1. INTRODUCTION

Route *choice models* play an important role in many transport applications and help to understand why people travel the way they do and to predict what they will do in the future. Route *choice set generation* is an essential part of route choice modeling in order to establish the weight of several route attributes in the decision process and to predict chosen routes in simulators. Route choice modeling for bicyclists is a topic of increasing interest as more and more people travel by bicycle for their daily commute, leading to problems with congestion in cycling lanes and at traffic lights as well as parking problems with bicycles. This in turn leads to traffic conflicts with both vehicles and pedestrians, creating unsafe situations. Understanding more about how and why cyclists travel and where they deviate from the shortest path, helps us to propose ways to improve safe cycling infrastructure and to subsequently study the effects of the modifications. Several attributes of a route are significant factors in the choice process: e.g. the number of left turns, the number of speed bumps, distance, slope, scenery etc. This study investigates the use of route complexity as an *additional* attribute. The *complexity* of a *given* (observed) path in a graph is the *minimum* number of shortest paths that is required to specify that path in the network. It can be interpreted as the (minimum) number of intermediate destinations that are connected by shortest subpaths. Note that *complexity* is a graph theoretical property and is not related to geometric properties of the route. *Complexity* is a path attribute which is considered to be important for route choice. The complexity was determined (i) for each route in a large set of routes observed by means of GPS traces and (ii) for routes in the choice sets for the origin-destination pairs corresponding to the observed routes generated by implementations of BFS_LE and DSCSG algorithms in the POSDAP tool by (ETH-Zurich, 2012). The distributions of observed routes and these two route choice generators seem to significantly differ. The complexity of the routes in the generated choice sets does not reflect the traveller behaviour we observed in the paths chosen by cyclists. This study looks at two route choice generation techniques and how they compare to the observed routes taken by bicyclists.

This chapter is organized as follows:Section ***Background*** briefly reviews the concept of choice set generation and various choice set generators that are described in the literature. Section ***Route complexity*** defines the concept of *route complexity* and describes an algorithm to compute the complexity for a given route. Section ***Case study*** describes the data set of chosen bicyclist routes, the distribution for the observed complexity and the relations between route properties. Section ***Discussion*** shows that the distributions for *route complexity* in generated choice sets significantly differ from the observed one.

8.2. BACKGROUND

Choice sets play a crucial role in route choice modeling and prediction. In choice set generation, the universal set U contains all possible routes from the origin to the destination. Such universal set can be infinitely large if it is allowed to include cycles (hence not only graph theoretical *paths* but also *walks*).

In *route based* choice models, finite choice sets are established. Each route in the choice set bears a collection of attributes (distance, number of junctions, scenery etc). A discrete choice model is used to predict the traveller's choice from the attributes. Most

models are based on multinomial logistic regression (MNL) and correction factors are introduced to account for correlation between overlapping routes. Model parameters and correction factors are determined using a finite choice set. Recursive logit (RL) models described by (Fosgerau et al., 2013) and by (Mai et al., 2015a) do not require a choice set for model *estimation*. Conceptually, they are equivalent to MNL models for route choice from an infinite number of alternatives. The model described in (Fosgerau et al., 2013) allows to compute the ratio of the probabilities of two routes due to the IIA (independence of irrelevant alternatives) property. RL uses link-additive attributes as opposed to route attributes and conceptually applies an MNL at each junction in order to predict the next link. Hence, it can be interpreted as a *link based* choice model. However, in order to *apply* route choice models in stochastic travel simulators, candidate routes need to be generated and compared also after estimating an RL model.

A typical choice set faced by a cyclist can include different paths with detours from the shortest path (i) to avoid dangerous situations such as busy highways, poor pavement conditions, unlighted cycle paths in the dark or unsafe neighborhoods or (ii) because of personal preference for certain areas like a park, slope, signalized junctions or a familiar path. Various choice set generators have been published.

(C. Prato & Bekhor, 2006) provides a method called Branch and Bound, which recursively constructs paths that satisfy specific conditions i.e. *directional, temporal, similarity, loop and movement* (avoiding left turns). For example with the temporal constraints, a route will only be included if its travel time is not larger than the shortest time multiplied by a given factor.

(Rieser-Schüssler et al., 2013) came up with a shortest path method, called Breadth First Search Link Elimination (BFS_LE). The BFS_LE method first computes the least cost path from origin to destination. Then links are eliminated in a particular order and a new shortest path is found. BFS refers to the fact that a tree of networks is considered and in each network a shortest path is determined using the A* algorithm. The tree is constructed by consecutively eliminating each element from the shortest path such that each recursively generated network differs in exactly one edge from the parent network in the recursion.

(Kazagli et al., 2016) introduced the concept of Mental Representation Items (MRI), to construct a data set they made use of a layer system. The first layer is used to determine a MRI choice set, such as $C_1 = \{avoidCC, aroundCC, throughCC\}$ where CC stands for the city center. A layer on top of that can provide additional details. In order to make the choice set operational, an attribute is assigned to each MRI by calculating the expected maximum utility, by taking the sums of the logarithms of all utilities on the path.

The Double Stochastic Generation Function method (DSCSG) described by (Nielsen, 2000) for public transportation by (Bovy & Fiorenzo-Catalano, 2007) produces heterogeneous routes because both the cost and parameters used in the cost function for the links are drawn from a probability function. A possible difficulty of this method is the high computational cost, however (Hood et al., 2011) shows DSCSG to be faster than the BFS_LE proposed by (Rieser-Schüssler et al., 2013). (Halldórsdóttir et al., 2014) shows that DSCSG has a high coverage level of replicating routes taken by bicyclists and that it performs well up to 10 kilometer. Furthermore (Bovy & Fiorenzo-Catalano, 2007) states

that the method guarantees, with high probability, that attractive routes are in the choice set, while unattractive routes are not.

In order to generate realistic predictions, the distribution for each route attribute in the choice set needs to comply with the corresponding distribution found in observed sets. This requirement related to route *complexity* is investigated in this chapter.

8.3. ROUTE COMPLEXITY

The complexity of a given path in a graph is the minimum number of *Basic Path Components (BPC)* in the decomposition of the path where a basic path component is defined as either a *least cost path* or a *non-least cost edge*. A non-least cost edge is an edge e whose vertices are connected by a path having a lower cost than the cost to traverse e.

Figure 8.1 shows the minimum decomposition for an observed sample path p (blue continuous line) in a graph having complexity $c(p) = 3$. The example shows that multiple decompositions do exist for path p.

(Knapen et al., 2016) define non-cyclic trips as *utilitarian* and formulate the hypothesis that in utilitarian trips, individuals tend to construct their routes as a concatenation of a small number of basic path components. Utilitarian trips have a purpose different from the fun of driving. They are driven with the intention to perform an activity at the destination location. (Knapen et al., 2016) present Algorithm 8.3.1 to determine the complexity of a path (i.e. the minimum number of basic path components).

Algorithm 8.3.1 Algorithm to determine the size of the minimum decomposition of a path into basic path components.

Input Graph G, Edge costs c, $P = (v_0, v_1, \ldots, v_l)$ containing no non-least-cost edges
$start \leftarrow 0$
$k \leftarrow 1$ ▷ k is the minimum decomposition size
while $P(v_{start}, v_l)$ is not a least cost path **do**
▷ Find the first vertex v_j in $P(v_{start}, v_l)$ such that $lc(v_{start}, v_j) < c(P(v_{start}, v_j))$
$v_j \leftarrow findFirstJoinVertex(P, v_{start})$
$k \leftarrow k+1$.
$v_{start} \leftarrow v_{j-1}$.
end while
return k

In algorithm 8.3.1 we have a graph G with positive edge costs c and a path $P = (v_0, v_1, \ldots, v_l)$ with no non-least-cost edges. Non-least-cost edges are easily determined in advance and each of them constitutes a BPC. Variable *start* is the index of the first vertex in a basic path component. Variable k is the minimum decomposition size. In the *while* loop we look for the first vertex v_j for which we can find a shorter path from v_{start} to vertex v_j; such vertices are called *join* vertices because in such vertex the given path and a shortcut *join* (see (Knapen et al., 2016) for details). In a *join* vertex we increment counter k by one. The predecessor of the join vertex is used to continue.

After the loop completes we can split the path at the vertex right before each *join* vertex, the vertex preceding a join vertex is called the split vertex. Using this algorithm, a

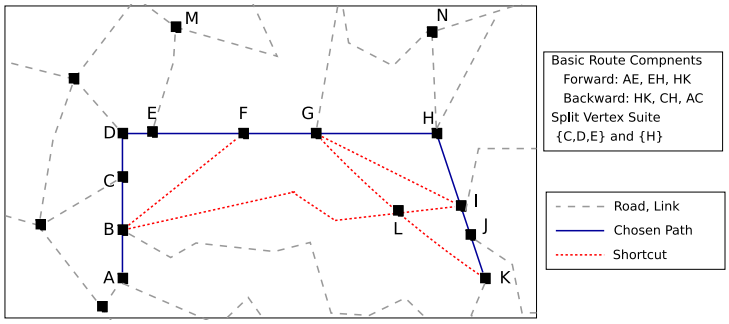


Figure 8.1: The blue continuous line visiting vertices A, B, C…I, J, K is the path followed by the traveler. Paths BF, BLI, GLI, GLK, etc represent shortcuts to the chosen path. There are two sets of split vertices: {C,D,E} and {H}. Hence there are three basic path components (BPC). Sample decompositions are ((A,C),(C,H),(H,K)) and (A,E),(E,H),(H,K)).

splitting is found at $k-1$ vertices, splitting our path P into k basic path components. (Knapen et al., 2016) showed that the decomposition is minimal but not necessarily unique. For example by running the algorithm in reverse direction of the path we may find a different but minimal decomposition by identifying *fork* vertices. The technique is illustrated by the example shown in Figure 8.1. The algorithm determines the least cost path from A to B and finds out that this coincides with the chosen path. This is repeated for the consecutive vertices and it turns out that the paths up to and including $\langle A,B,C,D,E\rangle$ are least cost paths but $\langle A,B,C,D,E,F\rangle$ is not. Hence, F is a join vertex: the lower cost path $\langle A,B,F$ *joins* the observed path in vertex F. The same algorithm is then applied again starting from vertex E (the predecessor of F in the observed path) and vertex I is found to be a join vertex too. No other joins are found in the *forward* pass.

A *backward* pass is the executed starting from the tail vertex of the observed path. Sub-paths $\langle J,K\rangle$, $\langle I,J,K\rangle$ and $\langle H,I,J,K\rangle$ are least-cost paths but $\langle G,H,I,J,K\rangle$ is not because $\langle G,I,J,K\rangle$ and $\langle G,L,K\rangle$ have a lower cost. Vertex G is a *fork* vertex. The set of vertices enclosed between a *fork* and its corresponding *join* is a called a *splitVertexSuite*. Figure 8.1 shows two *splitVertexSuite*s $\langle C,D,E\rangle$ and $\langle H\rangle$ respectively.

Knapen et al. (Knapen et al., 2016) prove (i) that the same numbers of *join* and *fork* vertices are found, (ii) that the vertices in the *splitVertexSuite*s are potential split vertices (i.e. intermediate destinations that the traveler may have had in mind) and (iii) that each minimum decomposition consists of exactly one vertex from each *splitVertexSuite*. Not every combination of *splitVertexSuite* members constitutes a valid decomposition. In (Knapen et al., 2017) the authors provide an algorithm to enumerate all valid decompositions. E.g. the set $\{C,H\}$ may constitute a valid decomposition generating three BPC: $\{\langle A,C\rangle,\langle C,H\rangle,\langle H,K\rangle\}$.

Figure 8.2 is taken from (Knapen et al., 2016) and shows the distribution for the complexity found in several data sets for which the majority (Belgian case) or all (Italian case) trips are car trips. This supports the hypothesis that utilitarian trips are composed of a

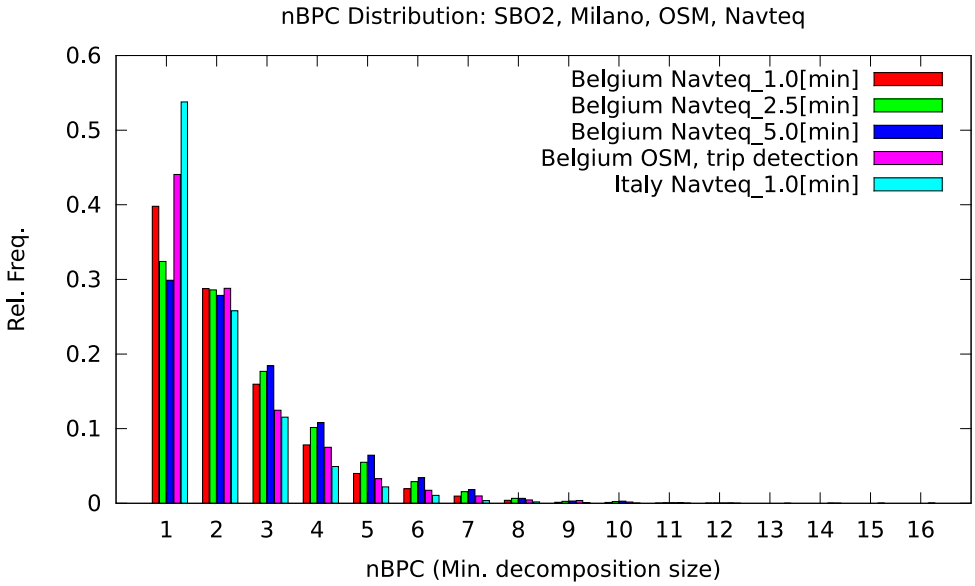


Figure 8.2: Relative frequency distribution for the size of the minimum decomposition of paths derived from GPS recordings. The Belgian set consists of *person* traces. It was map-matched using different networks and gap-filling thresholds. The Italian set consists of car traces only (recorded by on-board-unit (OBU).

small number of basic path components. Note that 95% of all trips had a complexity lower than 6 basic path components.

8.4. CASE STUDY

This study considers a trip to be utilitarian if and only if it does not contain a cycle and $r_d = d_{obs}/d_{short} \leq 1.08$ where d_{obs} and d_{short} are the observed and shortest route lengths respectively (details are found in (Wardenier et al., 2019)). This definition is stricter than the one used in (Knapen et al., 2016) in which all trips not containing any cycle have been considered.

8.4.1. COLLECTING DATA OF BICYCLE MOVEMENTS

The Dutch 2016 FietsTelWeek (Bike Counting Week) data set ((Bikeprint, 2017)) is available at http://www.bikeprint.nl/fietstelweek/. It contains 282,796 unique trips (although the corresponding infographic http://fietstelweek.nl/data/resultaten-fiets-telweek-bekend/ mentions 416,376 trips having a total distance of 1,786,147 kilometers). In order to anonymize the data, parts were stripped from the head an from the tail of the trips; the length of the stripped parts was randomly from the range [0,400] meters using a uniform distribution. Entire road network links were cut away. This process removes short trips and may explain why only 282,796 trips are found in the dataset. It was collected by 29,600 cyclists who voluntarily participated in a week-long survey to track their bicycle movements using a smart-phone app in the week of 19th of September 2016. The application ran in the

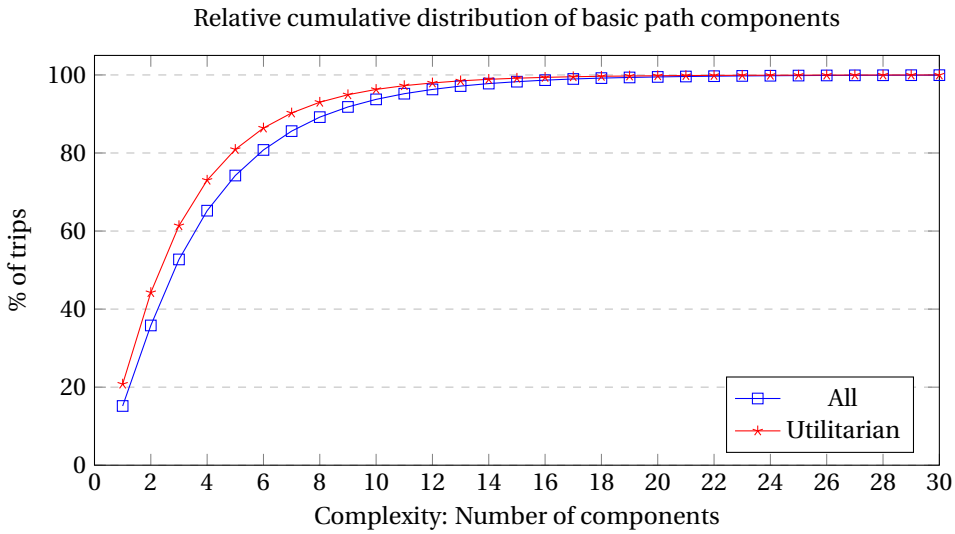


Figure 8.3: Cumulative distribution of the complexity of paths taken by bicyclists. Blue for unfiltered, red for only utilitarian trips with r_d <= 1.08.

background to collect the bicycle movements of all participants using the phone's GPS and acceleration sensors. The cyclists involved use their bike in a way often seen in The Netherlands to travel from and to work, supermarket, school, friends, etc. For privacy reasons, the resulting data was further anonymized in addition to the *head/tail stripping* mentioned above by the data provider before making it publicly available (i) by the removal of user information to make it impossible to trace multiple trips to a single person and (ii) by rounding of the trip departure time into one-hour bins to the nearest hour.

8.4.2. ROUTE COMPLEXITY IN REAL-LIFE GPS TRACES

The route complexity for the 282,796 collected by the Dutch FietsTelWeek2016 routes was computed and the distribution is shown in Figure 8.3 (blue line). For Flanders (Belgium) no detailed results for the *bike counting week* are made publicly available; hence, direct comparison is impossible. However, the distribution for the complexity of bicycle routes in The Netherlands significantly differs from the distribution for complexity found in *person traces* for Flanders shown in Figure 8.2. Car mode is the prevalent mode in Flanders according to the recurrent OVG travel behaviour survey . Hence most *person traces* consist of car trips and, as a consequence, most trips in the sets investigated by (Knapen et al., 2016) are car trips. The difference may result :

- from behavioral difference between car drivers an bicyclists,
- from regional behavior differences and
- from parameters chosen for the map-matching process because some map-matching algorithms fill gaps by connecting positions by the shortest path.

We had no control over the map-matching process because that was performed by the *FietsTelWeek* organizer. Access to raw GPS traces is required to exclude the latter possibility.

8.4.3. GENERATING ROUTE CHOICE SETS

To compare and analyze the conformance of reality, we looked at two route choice set generation methods: *Double Stochastic Generation Function (DSCSG)* by Halldorsdottir (Halldórsdóttir et al., 2014) and *Breadth First Search Link Elimination (BFS_LE)* by Rieser. (Rieser-Schüssler et al., 2013) and compared their output to the path complexity recorded in the Netherlands by the *FietsTelWeek* data-collection. For each observed trip, the origin and destination were extracted (OD-pair). We used an existing implementation of both algorithms in POSDAP (ETH-Zurich, 2012) to generate route choice sets for each OD-pair.

Only link length (travel distance) was used in the experiment. POSDAP allows to specify a set of link specific attribute values (like scenery, separate bike lanes etc): this was not used due to lack of data.

As there is no agreement on the size N_0 of the route choice sets, we arbitrarily state that the route choice generator should produce $N_0 = 16$ routes for each origin destination pair. The POSDAP software was slightly modified in order to execute at most a given number of $M = 128$ trials to find N_0 different routes (instead of running for a given duration) so that it behaves identically on different machines. For some origin destination pairs POSDAP is not able to find as many as N_0 routes in M trials, in which case we will use all found routes. The choice sets are written to CSV files for further processing.

We computed the complexity for each route in the choice set generated by POSDAP using the algorithm specified in (Knapen et al., 2016). The distribution of the path complexity was determined for the set of *predicted* paths (i.e. the paths in the generated choice sets).

8

8.5. DISCUSSION

8.5.1. RUN-TIMES

In terms of performance, BFS_LE is significantly quicker than DSCSG , producing 31,000 route-choice sets in 22 minutes for a instance with 6 parallel threads, averaging to approximately 248.3 choice set per minute per instance, on a machine with 2 *Intel Xeon CPU E5440* CPU's (4 cores/socket, 1 thread/core). DSCSG averaged to approximately 2.8 choice set per minute per instance on faster CPU's: 2 *Intel Xeon CPU E5-2660 v4* (14 cores/socket, 2 threads/core).

8.5.2. ROUTE COMPLEXITY IN GENERATED ROUTES

In figure 8.4 we plotted the different complexity distributions of the routes observed and of the choice sets generated by Double Stochastic Generation Function (DSCSG) and Bread First Search Link Elimination (BFS_LE) respectively. The results are in line with what we expected based on the nature of both algorithms. First we explain the distribution of BFS_LE as follows based on the structure of network in Amsterdam. A road network is said to be *dense* with regard to a set of observed routes if the average length

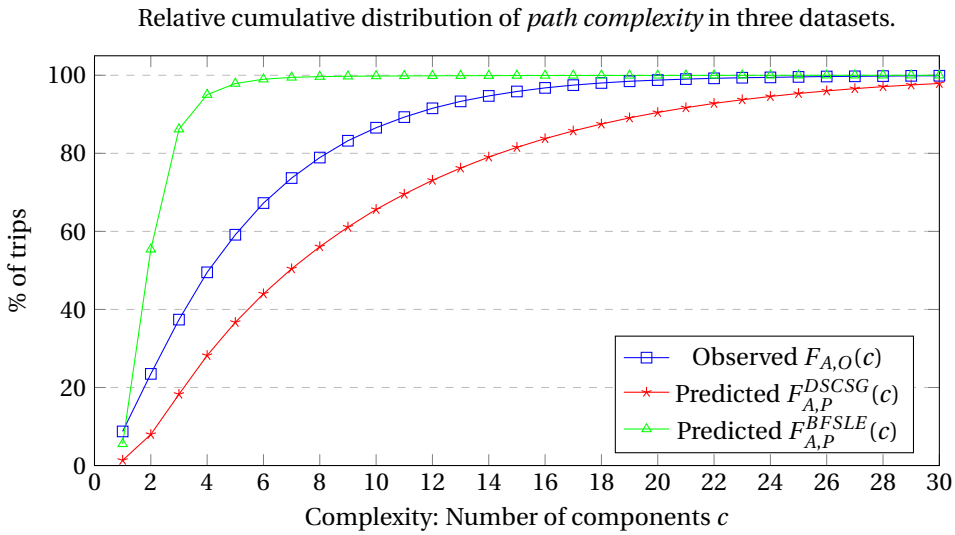


Figure 8.4: Cumulative distributions of number of basic path components of observed bicycling routes in the Amsterdam (blue) and the number of components in paths predicted by POSDAP 's implementations of Double Stochastic Generation Function (DSCSG) and Bread First Search Link Elimination (BFS_LE).

of network links is small relative to the developed length for the observed routes. Equivalently, a network has a high *density* if and only if commonly observed routes contain many links. In the observations for Amsterdam in the `fietstelweek2016` case, the network seems to be dense with regard to the set of shortest paths associated with each observed OD-pair. In most cases the shortest path $SP(o,d)$ for OD-pair $\langle o,d\rangle$ contains many links. This is shown in Figure 8.5 for the Amsterdam case. If the required size for the choice set for $\langle o,d\rangle$ is smaller than the number of road links in $SP(o,d)$, each route generated by BFS_LE is derived by finding the shortest path $SP_m(o,d)$ in a modified network where exactly one link link belonging to $SP(o,d)$ was removed. This is easily verified in Algorithm 8.5.1. Line 1 specifies the recursive BFS_LE procedure. *elimLinkSetsColl* is a collection of *link sets*. Each such link set in turn is used to eliminate links from the network.

Algorithm 8.5.1 BFS_LE algorithm.

```
Require: nPathsReqd, O, D, network
  function generate(elimLinkSetsColl, paths)
    elimLinkSetsCollNextLevel ← ∅
    for all elimLinkSet ∈ elimLinkSetsColl do
      network.removeLinks(elimLinkSet)
      sp ← shortestPath(O, D)
      if sp ∉ paths then                                  ▷ New one found
        paths ← paths ∪ {sp}
        if | paths | ≥ nPathsReqd then
          return
        else
          for all link ∈ sp do
            es ← elimLinkSet ∪ {link}
            if es ∉ elimLinkSetsCollNextLevel then
              elimLinkSetsCollNextLevel.add(es)
            end if
          end for
        end if
      end if
      network.addLinks(elimLinkSet)
    end for
    generate(elimLinkSetsCollNextLevel, paths)
  end function
  elimLinkSetsColl ← ∅
  paths ← ∅
  GENERATE(elimLinkSetsColl, paths)
```


The road network *density* (as defined above) severely affects the distribution for the route complexity in the choice sets generated by BFS_LE . Figure 8.5 shows the first part of the (fat tail) distribution for the number of links in the shortest path for each observed OD-pair. Only 9.4% of the shortest paths contain at most 16 links. The required choice set size is 16. Hence, in 90.6% of the cases the generated routes are derived from the shortest path by eliminating only one link (belonging to the shortest path) from the network.

In contrast to BFS_LE , DSCSG uses randomness to the cost function to generate new paths and thus subsequently the number of links in the shortest paths has less influence.

8.5.3. CHOICE SET QUALITY ASSESSMENT

In order to assess the quality of route choice sets, a technique to evaluate the similarity of two routes is required. Three methods are discussed in this section. All methods use the same concepts of *coverage* and *consistency* defined in Section 8.5.3 but use different similarity (overlap) functions.

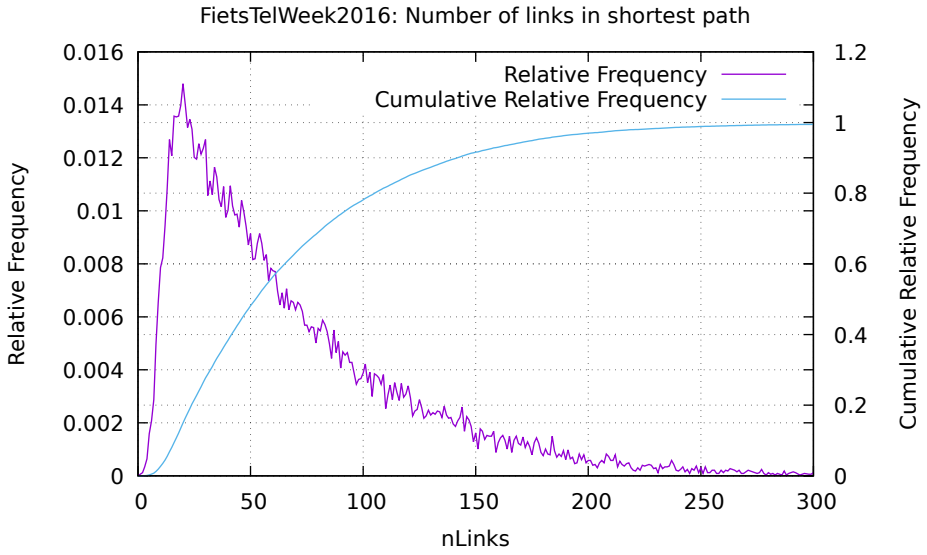


Figure 8.5: Distribution for the number of links in the shortest path linking O to D in the observed routes. Only 9.4% consists of at most 16 links.

COVERAGE BASED ON LINK MATCHING

In order to compare our routes generated with what we found in the literature, we computed *coverage (COV)* and behavioral *consistency (CON)* as found in Prato(C. Prato & Bekhor, 2007), Halldorsdottir (Halldórsdóttir et al., 2014) and others.

In general, multiple observed routes may share a single OD-pair. Let S^{odp} denote the set of OD-pairs and $S^{obs}(p)$ denote the set of observations for $p \in S^{odp}$.

The *coverage* realized by a particular choice set generator for a given set of observed routes is the fraction of observed routes for which the generated choice set contains *at least one* route for which the similarity to the observed one exceeds a given threshold.

Let $I(.)$ be the *coverage indicator function* which equals one in case its argument is true and zero otherwise. Let S^{OD} denote the set of observed OD-pairs. Let $O(r^{obs}, \rho)$ denote the fraction of overlap between the observed route r^{obs} and a route ρ in the choice set $S^{CS}(r^{obs})$ for r^{obs}. Let δ denote a threshold value. Then the coverage COV is defined by:

$$O(r^{obs}, \rho) = \frac{(length(overlap(r^{obs}, \rho))}{length(r^{obs})} \tag{8.1}$$

$$COV = \frac{1}{\mid S^{odp} \mid} \cdot \sum_{p \in S^{odp}} \left(\max_{r \in S^{obs}(p), \rho \in S^{CS}(p)} I(O(r, \rho) \geq \delta) \right) \tag{8.2}$$

Table 8.1: Coverage and Behavioral Consistency of Path Generation Techniques.

Path Generation Technique	Coverage (%) for overlap threshold				Behavioral Consistency
	100%	90%	80%	70%	
BFS_LE	14.58	28.96	38.37	47.89	0.642
DSCSG	11.01	22.80	33.26	44.69	0.645

Note that

$$\max_{\rho \in S^{CS}(r^{obs})} I(O(r^{obs}, \rho) \geq \delta) \in \{0, 1\} \tag{8.3}$$

$$COV \in [0, 1] \tag{8.4}$$

and that equation (8.3) was used to speed up the computation.

The *index of behavior consistency (CON)* considers each observed route r and compares it to *all* routes in the appropriate choice set in order to find the largest similarity value (i.e. finding the best match). The *index of behavior consistency (CON)* is the average value, computed over all observations, of the maximal similarity between the observed route and *any* of the generated routes.

The *index of behavior consistency (CON)* measure compares a path generation method with an algorithm that would replicate *all* observations. It is formally defined by

$$CON = \frac{1}{\mid S^{odp} \mid} \cdot \sum_{p \in S^{odp}} \left(\frac{1}{\mid S^{obs}(p) \mid} \cdot \sum_{r \in S^{obs}(p)} \left(\max_{\rho \in S^{CS}(p)} O(r, \rho) \right) \right) \tag{8.5}$$

where: CON is the consistency index and $O(r, \rho)$ is the overlap between the routes r and ρ (see Equation (8.1)).

In the experiment using `fietstelweek2016` data, each OD-pair has exactly one observation. In that case the expressions for *coverage* and *consistency* reduce to

$$COV = \frac{1}{\mid S^{obs} \mid} \cdot \sum_{r \in S^{obs}} \left(\max_{\rho \in S^{CS}(r)} I(O(r, \rho) \geq \delta) \right) \tag{8.6}$$

$$CON = \frac{1}{\mid S^{obs} \mid} \cdot \sum_{r \in S^{obs}} \left(\max_{\rho \in S^{CS}(r)} O(r, \rho) \right) \tag{8.7}$$

What we see in low values for coverage and behavioral consistency in table 8.1 is similar to what we see in the route complexity distributions in Figure 8.4: the predicted routes have a low conformance to reality.

COVERAGE BASED ON GEOMETRIC DISTANCE

Different *geometric distances* are proposed as alternatives for assessment of similarity by *matching complete links.* Hausdorff and Fréchet distance are evaluated as similarity measures to compare routes.

Firstly we looked at Hausdorff distance between the observed and each generated path, which verifies whether every point on the route is close to the some other point

in the other route. The Hausdorff distance is the longest distance between any point p_0 on the observed route to the point p_1 on the generated route that is nearest to p_0. We use the euclidean distance $d_E(.,.)$ between points as the base metric to define the Hausdorff distance $d_H(.,.)$ between routes as follows:

$$d_H(r_0, r_1) = \max\Big(\max_{p_0 \in r_0} \min_{p_1 \in r_1} d_E(p_0, p_1), \max_{p_1 \in r_1} \min_{p_0 \in r_0} d_E(p_0, p_1)\Big) \tag{8.8}$$

	min	median	mean	max	std dev
BFS_LE	0	93.491	297.009	25372.660	622.889
DSCSG	0	66.997	232.946	25372.660	584.124

Table 8.2: Statistics for the Hausdorff distance between an observation and the route with lowest Hausdorff distance in the corresponding choice set.

Table 8.2 shows that in general the minimum Hausdorff distance between an observed route and any of the routes in its associated choice set is smaller for DSCSG than for BFS_LE .

On the other hand the Fréchet distance measures the similarity between curves, taking into account the location and order of points along the curves, which gives more useful information about similarity. For example if we take a route and shuffle the order, that route will still have the same Hausdorff distance however the Fréchet distance will signal the difference.
Imagine two bicyclists cycling across town from the same origin to the same destination but over two different paths, the Fréchet distance would be equal to the shortest rope that would necessary to connect both cyclists. For example if the two bicyclists would take a nearly parallel path but the first bicyclist on the left bank of a river and the other bicyclist on the right bank of a winding river, the coverage and consistency would be nearly 0 but the Fréchet distance would be limited to the width of the river.

	min	median	mean	max	std dev
BFS_LE	0	84.904	243.549	25897.228	583.795
DSCSG	0	107.04	306.603	25897.228	624.756

Table 8.3: Statistics for the minimum Fréchet distance between an observation and the routes in the corresponding choice set.

Table 8.3 lists the Fréchet distances for both generation methods. It shows that BFS_LE on average is able to produce routes more similar to the observed one than DSCSG does. This is opposite to the evaluation by Hausdorff distance, meaning that routes generated

with DSCSG might be Hausdorff-closer to the observed routes but when taking into account curves with Fréchet the routes generated with BFS_LE have a lower Fréchet distance.

COVERAGE BASED ON GEOMETRY BUFFERING

Additionally we looked at a different way to measure overlap and coverage, by allowing a given margin around one of the routes in the comparison.

The GIS (Geographical Information System) *buffer* concept is used. A buffer with radius R around a geometry g is a geometry $b(g,R)$ that contains all points at a distance from g that is less than or equal to R.

In order to evaluate the *nearness* of a route r_0 to route r_1 we define the buffer $b(r_1,R)$. We transform the concatenation of the geometries of all links in the route r_0 into a single line-string which is then subdivided into patches of a predefined length (except for the last one). A patch s_i in the reconstitution of route r_0 is counted to be *near to* r_1 if and only if $s_i \cap b(r_1,R)$ (the patch intersects the buffer). In case of intersection, the total length of the patch is assumed to be *near* r_1. Let ρ_0 denote the reconstitution of r_0. The similarity is defined as:

$$sim(r_0,r_1) = \frac{\sum_{s\in\rho_o | s\cap b(r_1,R)\neq\emptyset} len(s)}{\sum_{s\in\rho_0} len(s)} \tag{8.9}$$

When evaluating choice sets, the *observed* route is subdivided into short patches and the buffers are defined around the *generated* routes.

By dividing up the complete geometry of the observed route into small patches of d meter, we can use GIS to determine whether any point on each patch intersects with the generated path. The total distance of all patches on the observed path that intersect the generated path, divided by the total length of the observed path gives us a overlap fraction. that is used to determine an approximation for the observed route covered by the buffer. The fraction of the observed route length covered is used as a quality measure. The smaller we make d patch size, the higher accuracy we get as we define intersection as two geometries touching each other at all possible points. However this is a trade-off with computation time as the work load multiplies, in this study we used a patch size of 50 and 5 meters.

By defining a *buffer* around the geometry of each observed route we can apply a *spatial tolerance* while comparing two paths (link sequences). If a path p_0 is within the buffer $b(p_1,r)$ we consider paths p_0 and p_1 to overlap (and hence to be equivalent). For example: the prediction p_{pred} follows a path parallel to the observation p_{obs} but over the cycle-way on the other side of the road.

For each observed route U we have the generated choice set $CS(U)$ that contains $N(U)$ routes. For each generated route in $CS(U)$ we determine a buffer with radius R meters and look at the total length of the patches on the observed routes that intersect with the buffer of the generated path. For each observed route and for both route choice set generation methods (BFS_LE and DSCSG) we determine the generated path with the highest overlap fraction. The corresponding maximum overlap is called

	50 meter patches		5 meter patches	
	BFS_LE	DSCSG	BFS_LE	DSCSG
Using 1 meter buffer	0.696	0.711	0.665	0.662
Using 5 meter buffer	0.710	0.733	0.681	0.690
Using 50 meter buffer	0.788	0.842	0.770	0.817
Using 250 meter buffer	0.871	0.912	0.858	0.900
Using 500 meter buffer	0.916	0.944	0.914	0.914
Using 1000 meter buffer	0.947	0.963	0.956	0.969
Using 2500 meter buffer	0.964	0.973	0.980	0.983

Table 8.4: Average of $maxOverlap(U, CS(U))$ over all observed routes for two different resolutions (5 and 50 meters). A value of 1 indicates that *each* choice set contains *at least one* route that *coincides* (using a given distance threshold) with the observed route). A value of 0 indicates that for all choice sets there is no route that has *any* link near to the observed route (because we consider *maxOverlap*$(U, CS(U))$).

	50 meter patches		5 meter patches	
	BFS_LE	DSCSG	BFS_LE	DSCSG
Using 1 meter buffer	0.674	0.701	0.646	0.653
Using 5 meter buffer	0.689	0.722	0.661	0.680
Using 50 meter buffer	0.767	0.825	0.748	0.806
Using 250 meter buffer	0.846	0.894	0.834	0.884
Using 500 meter buffer	0.892	0.925	0.889	0.923
Using 1000 meter buffer	0.923	0.944	0.932	0.951
Using 2500 meter buffer	0.940	0.953	0.959	0.966

Table 8.5: Average of $maxOverlap(U, CS(U))$ over all observed routes for two different resolutions (5 and 50 meters). A value of 1 indicates that *each* choice set contains *at least one* route that *coincides* (using a given distance threshold) with the observed route). A value of 0 indicates that for all choice sets there is no route that has *any* link near to the observed route (because we consider *maxOverlap*$(U, CS(U))$).

$maxOverlap(U, CS(U))$ and is used as a quality measure for the choice set $CS(U)$. Table 8.5 lists the average (over all observations U) of the $maxOverlap(U, CS(U))$ for both choice set generation methods (BFS_LE and DSCSG). We experimented using two different patch sizes of 50 and 5 meters respectively to evaluate the effect of the resolution on the precision; as expected we see a lower maximum overlap when using a smaller patch size since we are able to look at a closer level at the distance between generated and observed path. Intersection of the observed route with a 5 meter patch S_5 implies intersection of the observed route with each 50 meter patch S_{50} that contains S_5 but the inverse is not true.

Tables 8.6 to 8.12 show coverage values calculated in a similar way as in table 8.1 but using different *overlap* functions. We used a large range of buffer radius values in order to verify to what extent the difference between the generated and observed routes was limited to being on the different side of the street, river, city block or neighbourhood. For 1[m] and 5[m] buffers, BFS_LE has a larger coverage than DSCSG in most cases.

Path Generation Technique	Coverage (%) for overlap threshold				Behavioral Consistency
	100%	90%	80%	70%	
BFS_LE	11.50	29.59	39.06	48.62	0.646
DSCSG	8.71	23.69	34.52	46.19	0.654

Table 8.6: Coverage with 1 meter buffer.

Path Generation Technique	Coverage (%) for overlap threshold				Behavioral Consistency
	100%	90%	80%	70%	
BFS_LE	12.06	31.68	41.40	51.05	0.661
DSCSG	9.65	27.04	38.71	51.31	0.680

Table 8.7: Coverage with 5 meter buffer.

Path Generation Technique	Coverage (%) for overlap threshold				Behavioral Consistency
	100%	90%	80%	70%	
BFS_LE	12.77	48.62	56.57	63.79	0.748
DSCSG	13.47	54.12	63.77	72.23	0.806

Table 8.8: Coverage with 50 meter buffer.

Path Generation Technique	Coverage (%) for overlap threshold				Behavioral Consistency
	100%	90%	80%	70%	
BFS_LE	13.32	64.00	71.44	76.95	0.834
DSCSG	14.11	72.21	79.73	85.29	0.884

Table 8.9: Coverage with 250 meter buffer.

Path Generation Technique	Coverage (%) for overlap threshold				Behavioral Consistency
	100%	90%	80%	70%	
BFS_LE	13.49	73.95	82.02	86.46	0.889
DSCSG	14.27	81.65	88.10	91.45	0.923

Table 8.10: Coverage with 500 meter buffer.

Path Generation Technique	Coverage (%) for overlap threshold				Behavioral Consistency
	100%	90%	80%	70%	
BFS_LE	13.67	82.93	90.47	93.54	0.932
DSCSG	14.52	88.38	93.48	95.75	0.951

Table 8.11: Coverage with 1000 meter buffer.

Path Generation Technique	Coverage (%) for overlap threshold				Behavioral Consistency
	100%	90%	80%	70%	
BFS_LE	13.71	89.51	95.87	97.84	0.959
DSCSG	14.63	92.09	96.35	98.01	0.966

Table 8.12: Coverage with 2500 meter buffer.

Starting at 50[m] BFS_LE consistently reports a lower coverage than DSCSG .

Using larger *patches* measures coincidence of sub-routes less accurately than using smaller patches; furthermore, the coincidence is consistently overestimated. This is because a large patch indicates large overlap length even if only a single point of the observed route is near to the generated one.

The over-estimation grows with the *patch* length. In cases where the observed and generated routes *share* some nodes but the sub-routes connecting consecutive shared nodes are spatially not near to each other, the use of longer patches generates a larger over-estimation. Therefor, the coverage increases with the buffer radius (explaining difference between tables).

Regarding the differences within the tables, observe that the ratio $COV_{bfsle}/COV_{dscsg} < 1$ for buffer radius values up to 5[m] and $COV_{bfsle}/COV_{dscsg} > 1$ for the other cases. This reveals that the shortest routes selected by BFS_LE (which are not necessarily the *K shortest paths*) on average have less points near to the observed route than the (possibly more complicated) routes generated by DSCSG .

Table 8.1 is taken as the reference case. It represents the case where network links need to be replicated entirely (*overlap* is equivalent to *identification*). In the buffer based methods, *overlap* is defined in terms of *geometric neighborhood*: this leads to a relaxed requirement (identification is not a necessary condition) and hence, for each given choice set, the *coverage* and *consistency* quality values can be expected to grow with the buffer radius used in the overlap function.

This phenomenon is explained as follows. The coverage indicator function used in Equation (8.2) uses *overlap* as a similarity measure (see Equation (8.1)). The *overlap* function used to generate Table 8.1 considers *entire links*. In the case of Table 8.6, the overlap function (i) considers *patches* (limited length parts of the links in the observed route) and (ii) considers a predicted route to cover a *patch* if it is *sufficiently nearby* (i.e. coincidence is not even required).

Figure 8.6: Generated paths by BFS_LE , red-dashed path indicates the observed route.

DISTANCE MEASURES AND QUALITY ASSESSMENT

Several measures for dissimilarity between routes have been discussed in previous sections. These are based on euclidean distance and hence on the notion of nearness in space.

All of them can be used a quality measures for a choice set.

Assume an OD-pair $\langle O, D\rangle$, its associated choice set $C(\langle O, D\rangle)$ and a set $S^{obs}(\langle O, D\rangle)$ of observed routes for the OD-pair. The choice $C(\langle O, D\rangle)$ set is of high quality *only if* for each observed route $r^{obs} \in S^{obs}$ it contains a route r having a small dissimilarity $d_X(r, r^{obs})$. This is a *necessary but not sufficient* condition for quality.

Another requirement for high quality of a choice set is *diversity* of the routes with regard to their attributes that are assumed relevant in the choice process. Such attributes may be rated differently by different categories of travellers. Insufficient variation makes it harder to understand why a particular participant took the observed route. This variety may require routes not similar to the observed one in terms of the evaluation methods presented above.

An additional quality measure is the compliance of the distribution for route complexity (discussed in Section 8.3) in the choice set with the one observed in reality (discussed in Section 8.4.2).

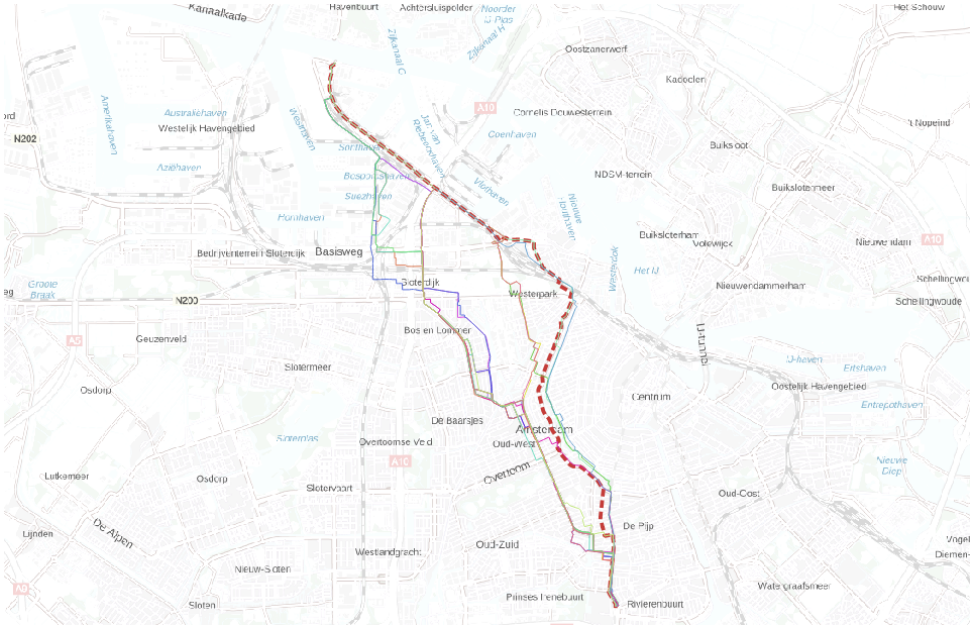


Figure 8.7: Generated paths by DSCSG , red-dashed path indicates the observed route.

8.6. CONCLUSION

There are various methods to generate route choice sets. In this chapter we used two of them. Double Stochastic Generation Function (DSCSG) is evaluated because it generates heterogeneous routes, because it is reported to perform well for trips up to a length of 10 kilometers and because it puts more attractive routes in the choice set. The problem with this kind of route choice generation is that the generated route can be overcomplicated and unrealistic. Secondly we used Breadth First Search Link Elimination (BFS_LE) to compare run-times and output.

This study formally defines the concept of *route complexity* and computes complexity distributions for both a set of observed routes by bicyclists and a set of routes generated by the POSDAP implementations of BFS_LE and DSCSG . The distributions of the generated paths are shown to be significantly different from reality.

We propose two options to solve the problem. Firstly we could attempt to find a technique where we integrate route complexity inside route choice generation algorithms such as BFS_LE and DSCSG in order to filter out the most inappropriate routes. However this may turn out to be impossible since the quality criterion applies to the distribution of a particular route property as opposed to a route specific property. A second option consists of a two-step method that first applies classical route choice-set generators like BFS_LE and DSCSG to generate an initial choice-set that we subsequently reduce in order to make sure the complexity distribution is more realistic, as described in prior work by (Wardenier et al., 2019). Finally, resulting choice-sets can be evaluated by their complexity distribution and by their coverage and consistency quality indicators making use of the proposed relaxed similarity functions as in this chapter.

9

LIMITATIONS OF RECURSIVE LOGIT FOR INVERSE REINFORCEMENT LEARNING OF BICYCLE ROUTE CHOICE BEHAVIOR IN AMSTERDAM

Used for route choice modelling by the transportation research community, recursive logit is a form of inverse reinforcement learning. By solving a large-scale system of linear equations recursive logit allows estimation of an optimal (negative) reward function in a computationally efficient way that performs for large networks and a large number of observations. In this chapter we review examples of recursive logit and inverse reinforcement learning models applied to real world GPS travel trajectories and explore some of the challenges in modeling bicycle route choice in the city of Amsterdam using recursive logit as compared to a simple baseline multinomial logit model with environmental variables. We discuss conceptual, computational, numerical and statistical issues that we encountered and conclude with recommendation for further research. Finally, we compare this work to a conventional choice model, estimating models using multi-nominal logit, mixed logit and mixed path size logit specifications. Our results show that cyclists have a highly stochastic behavior that are likely to prefer detours to drive over cycle-way infrastructure, near greener land-use and near water, and on less busy roads. Models such as mixed logit that can estimate the stochasticity of cyclists perform best to capture this behavior.

Parts of this chapter have been presented at the 100th Annual Meeting of the Transportation Research Board and the 3rd ACM SIGSPATIAL International Workshop on GeoSpatial Simulation. A third extension was presented at the 14th ACM SIGSPATIAL International Workshop on Computational Transportation Science (IWCTS 2021).

Bicycling in Amsterdam is serious business: one third of the daily movements in Amsterdam by residents and visitors is done by bike and almost half of the commute trips between work and home is done on a bike. Furthermore bicycling has seen a steady growth, not just in Amsterdam but in all (major) cities in the Netherlands. This leads to increasing congestion issues for bicyclists especially at intersections with traffic lights. To address this, policy changes are required such as new infrastructure, changes in traffic signal policies, etc. To create effective policies, policy makers need tools to gain insights in the behaviour of bicyclists, to model when and where they will bicycle.

In our previous research in (Koch et al., 2019) found that bicyclists in Amsterdam rarely take the shortest path and make more detours than cars. It is thus less straightforward to predict routes taken by the average bicyclist route and consequently estimate the number of bicyclists on a given street. However with the advent of new data collections techniques such as GPS on smartphones, more and more data is available on the revealed preference of bicyclists, allowing researchers to develop new ways to model influences on bicycle route choice. In this study we will to attempt to quantify the influence of environmental and spatial planning features on route choice decisions of everyday bicyclists in Amsterdam by estimating a model that can predict where and how these cyclists traverse in and around Amsterdam. We begin this chapter with a background reviewing literature relevant to bicycle route choice, by discrete choice modeling with generation of alternative routes. Next we review methods to model route choice without generating choice sets: recursive logit and inverse reinforcement learning. The case study describes the data set of observed bicycle routes. We also defined a number of environmental variables that potentially could have an effect on bicycle route choice. First we estimate a baseline multinomial logit model with the environmental variables, based on the observed routes compared with up to 16 generated route alternatives. Subsequently we describe our efforts to estimate a recursive logit model with the observed bicycle routes in our data, environmental explanatory variables, and the Amsterdam street network – without generating route choice alternatives. We discuss conceptual, computational, numerical and statistical issues that we encountered. We conclude the chapter with a reflection on our experience of modeling route choice of bicyclist in Amsterdam and recommendations for further research.

9.1. BACKGROUND

9.1.1. DISCRETE CHOICE MODELING OF TRAVEL ROUTES

Since the 1970's discrete choice modeling has been a leading method to understand choice behaviour of individuals in a wide range fields such as marketing, economics and transportation. Described by (McFadden, 1973) in 1973, discrete choice modeling has subsequently been extended over the decades in order to overcome specific limitations such as overlapping alternatives and correlations over time. The study of the specific field of route choice, is more complicated than a choice between easily enumerable distinct alternatives, as route choice is typically a sequence of choices at each intersection, each transit stop, each mode, etc. This leads to very large choice set that is theoretically infinite due to loops. Often there can also be a large overlap between different route alternatives leading to difficulties for choice modeling. We will highlight two com-

monly used approaches: An approach established in the 1990s to model route choice using a collection of observed paths and for each observations a set of generated paths by a route choice generator. This approach has been used to estimate models such as multi-nominal logit (MNL) and mixed logit. This approach comes with limitations: as discussed in (Koch et al., 2019), these route choice generators do not necessarily create realistic routes; and (Frejinger et al., 2009) argues that parameter estimates can vary significantly depending on the bias of the route choice generator. To address the issues with the overlap between difference alternative paths and the resulting correlations, multiple extensions have been proposed to attempt to avoid erroneous path probabilities and substitution patterns. The most two popular are path size logit (Ben-Akiva & Bierlaire, 1999) and C-Logit (Cascetta et al., 1996), which decrease the utility of overlapping paths proportional to the overlap with other paths included in the choice set.

A second approach is to achieve a consistent choice set by sampling as proposed by (Frejinger et al., 2009). This approach attempts to set up a sampling protocol in order to obtain unbiased parameter estimates from the route choice sets to neutralize the bias introduced by the route choice generator.

9.1.2. BICYCLE ROUTE CHOICE

In 2010, (Menghini et al., 2010) published a seminal route choice model for bicyclists estimated from a large sample of GPS observations in a revealed preference study with 2435 persons logging 73,493 trips in Zurich, Switzerland. Using this data they estimated a multinominal logit model and used breadth-first search link elimination (BFS-LE) to generate choice alternatives for each observed trip. They included six different variables in the choice model: length of the route, average absolute gradient change, maximum gradient change, percentage of marked bicycle paths along the route, number of traffic lights and the path size measure. Accounting for the similarity between alternatives with the path size vector, their model showed that the elasticity with respect to trip length was nearly four times larger than that with respect to the percentage of bicycle paths along the route. The only other explanatory variable that had an impact albeit small, was the product of length and the maximum gradient along the route.

(Sener et al., 2009) in 2009 estimated a choice model using a stated preference survey on Texas bicyclists and analyzed a comprehensive set of attributes that influence bicycle route choice, such as the bicyclists characteristics, on-street parking, bicycle facility types and amenities, roadway physical characteristics, roadway functional characteristics, and roadway operational characteristics. To estimate the model they used a panel mixed multinomial logit model. Their results indicate that travel time (for commuters) and motorized traffic volume are the most important attributes in bicycle route choice. Other attributes with a high impact include number of stop signs, red light, and cross-streets, speed limits, on-street parking characteristics, and whether there exists a continuous bicycle facility on the route.

(Hood et al., 2011) estimated a route choice model on GPS data collected by cyclists with a smartphone in San Francisco. Their choice set was generated by double stochastic method by (Bovy & Fiorenzo-Catalano, 2007). They avoid the issues with the overlap between different alternative paths by opting for a Path Size Multinomial Logit model and there results showed that bicycle lanes were preferred to other facility types, that

steep slopes were disfavored. Other negative attributes were length and turns. Traffic volume, traffic speed, number of lanes, crime rates, and nightfall had no effect.

(Broach et al., 2012) looked at are revealed preference dataset collected by 164 cyclists using GPS units. (Broach et al., 2012) used 1449 utilitarian trips to estimated a bicycle route choice model. To generate the choice-set they used an algorithm based on multiple permutations of path attributes and formulated to account for overlapping route alternatives. Their results from the Path Size Logit model suggest that bicyclists are sensitive to the effects of distance, turn frequency, slope, number of traffic signals, and traffic volumes. Additionally, bicyclists appear to highly value off-street bike paths, enhanced neighborhood cycleways with traffic calming features, and bridge facilities. Bike lanes more or less exactly offset the negative effects of adjacent traffic.

In 2017, (Ton et al., 2017) reported on a route choice model for bicyclists using the same dataset as in the study in this paper. Ton et al. consider the construction of choice sets via an empirical approach, using only the observed trips in the data set to compose the choice alternatives. On basis of their specific focus (inner city travel in Amsterdam) Ton et al. selected 6 variables: distance, percentage of separate cycle paths, number of intersections, rain, sunset and sunrise times and trip purpose. Their findings suggest that bicyclists in Amsterdam are insensitive to dedicated cycle paths, attributed to an inner city characterized by a dense road network where cycling is the most prominent mode of transport. Additionally they found that cyclists in Amsterdam were found to minimize travel distance and the number of intersections per kilometer. Furthermore they found that for early morning trips there was a stronger impact of distance on route choice than outside these hours. In a subsequent paper (Ton et al., 2018) looked at a data-driven path identification approach, combining all unique routes observed for one origin-destination pair into a choice set and comparing this approach with two commonly used choice set generation methods (breadth-first search on link elimination and labelling).

(Ghanayim & Bekhor, 2018) analyzed bicycle route choice for commuter trips using a dataset from a GPS-assisted household travel survey in the Tel Aviv metropolitan area. Their results indicate an expected tendency to ride in longer routes, but on separated bike lanes. In the absence of such lanes, riders prefer to use local streets and avoid busy arterial streets and highways. Their route choice generation calculated 20 alternatives using 3 methods: link elimination, link penalties and a simulation method that calculated a shortest path using link impedance after each draw from a log normal distribution. The paper estimated a choice model using 3 model forms: multinomial logit, C-logit and Path Size-logit.

(Bernardi et al., 2018) analyzed the GPS traces recorded by 280 cyclists in a mobility panel throughout the Netherlands, that made approximately 3500 bike trips over a four week period in 2014. The choice sets were composed by the shortest-path for the origin destination and 4 alternative paths that were observed for that origin-destination pair. Route choice models were estimated using a binomial logit model and a mix multinomial logit model with path size logit formulation. Their results show that trip lengths and trip distribution over time reveal a population sample much used to cycling, frequently and over long distances.

(Zimmermann et al., 2017) showed that is possible to estimate bicycle route choice

without the restrictiveness of pre-generated route choice sets and model route choice as a sequence of choices via recursive logit. For comparison with the recursive logit model, we will estimate a simple baseline multinomial logit model using a synthetically generated choice set. The synthetic approach allows us to generate additional plausible route alternatives outside the set of observed routes. This means that we can include all observations even between origin destinations pairs with only a single observations, unlike the study by (Ton et al., 2017) that is limited to trips traversing the inner city of Amsterdam, due to a insufficient density of trips in the suburbs for this empirical approach to work there. (P. Chen et al., 2018) examines the effects of built environment features, including factors of land use and road network, on bicyclists route preferences using GPS datasets collected in the city of Seattle. The choice model was estimated using a path-size-based mixed logit model. (P. Chen et al., 2018) identifies five core factors that influence route choice behavior: trip length, speed limit, slope, bicycle lanes, and street lights.

The study by (C. G. Prato et al., 2018) focused on observing bicyclist behavior in the cycling oriented country of Denmark, exploiting rich data about the cycling environment, estimating the model in value of distance rather than preference space. (C. G. Prato et al., 2018) not only focused on preferences for traditional variables such as distance, hilliness and road characteristics. but also on aspects such as bicycle facilities and land-use designations. They estimated a model on 3384 cycling trips using mixed path size logit.

(Dane et al., 2019) looked at the route choice behavior of cyclists on e-bikes based on 17626 trips from 742 users extracted from GPS data. In this paper a mixed logit model with addition of the path-size attribute is applied on the route choice of respondents. Choice sets were generated using the K-shortest path algorithm.

In our work (Koch et al., 2019) we found that bicyclists in Amsterdam often deviate from the shortest path, more than car drivers, indicating that there are different and possibly also more factors that have an effect on the routes bicyclists in Amsterdam take. In (Koch et al., 2019) we focused on the concept of route complexity: counting the number of locations where people deviate from the shortest path, in the interest of improving route choice generation techniques and potentially get more insight into the motivations for the route choice for bicyclists. In this study we explore other effects on route choice using different methodologies, without looking at route complexity or where people deviate from the shortest path. In future research we intend to combine both streams of work.

9.2. METHODS

In this section we will review two variants of performing choice modeling without choice sets: an analytical solutions via recursive logit and a computational approximation via inverse reinforcement learning.

9.2.1. DYNAMIC DISCRETE CHOICE MODELING OF TRAVEL LINK SEQUENCES

An alternative approach uses link-based Markov decision process to model route choice as a series of sequential decisions. First proposed by (Fosgerau et al., 2013) it uses a linear

system of equations to efficiently compute choice probabilities by using a solver to solve Bellman equations.

An incidence matrix is established that defines the exponential utility to perform action a from state k:

$$M_{ka} = \begin{Bmatrix} \delta(a|k)\, e^{\frac{1}{\mu} v(a|k)}, & a \in A(k) \\ 0 & \text{otherwise.} \end{Bmatrix} \tag{9.1}$$

The incidence matrix contains all the possible transitions from each state to another state or a terminating state (dummy link). Each dummy link d represents a link to a termination state for the last state in each observed trajectory. This makes the size of the incidence matrix $|\widetilde{A}|$: the sum of the the number of states A and the number of dummy links d. As the dummy links d have no successors, the row $k = d$ will be zero.

Secondly (Fosgerau et al., 2013) define a vector z of size $|\widetilde{A}x1|$ vector where $z_k = e^{\frac{1}{\mu} V(K)}$ and a vector b of size $|\widetilde{A}x1|$ where $b_k = 0, k \neq d$ and $b_d = 1$. Now given the identity matrix I, (Fosgerau et al., 2013) write the linear equation:

$$z = Mz + b \Longleftrightarrow (I - M)z = b \tag{9.2}$$

This system has a solution if $I - M$ is invertible, which might not be the case. As (Fosgerau et al., 2013) note this is highly dependent on the balance between the number of paths that connect the nodes in the network and the size of instantaneous utilities $\frac{1}{\mu} v(a|k)$. They note that this issue is particularly important to consider when estimating a model, as depending on the value of β, $I - M$ can be ill-conditioned or even singular. (Fosgerau et al., 2013) note that this limits the possible values of parameters, as when equation 9.2 does not yield a valid solution for at least one observation, the log likelihood function is not defined. They suggest to deal with this issue by starting at a feasible point (meaning a large enough magnitude in the parameters) and then being conservative in the initial step size of the line search algorithm at the price of an increased number of iterations.

(Mai et al., 2015b) proposed a nested recursive logit that relaxes the independence from irrelevant alternatives property of the logit model by allowing scale parameters to be link specific. (Zimmermann et al., 2017) subsequently look at bicycle route choice problem in the city of Eugene, Oregon. By using 648 observations of bike trips collected from 103 users. They test a long list of 14 potential parameters: length; link constant to penalize paths with many constants; length interacted separately with upslope, medium traffic, heavy traffic, regional multi-use path, bicycle boulevard, bike lane; bridge; bridge interacted with bike facilities; no turn; no turn interacted with crossroad; left turn interacted with crossroad separately for medium traffic and for heavy traffic.

In (Mai et al., 2018) an improvement is proposed to (Fosgerau et al., 2013) by reducing the numbers of linear systems that need to be solved. By adding all observed destinations in vector b of size $|\widetilde{A}x|D||$ it becomes possible to solve the problem one iteration instead of solving the system for each destination separately, allowing for 30 times performance gain in their example. They use this performance gain to propose a mixed recursive logit, which allows for random taste variation by adding a random value to the utility function and running the model n draws each iteration to allow for a random variation. They perform a case study in two cities. First a car route choice model

in the Swedish city of Borlänge, with 466 destinations, 1832 observations and a bicycle route choice model in Eugene, Oregon with 286 destinations with a unknown number of observations.

In (de Freitas et al., 2019), recursive logit is used to model inter-modal travel based on a static network that describes various connections in Zurich, Switzerland. The street network consists of 30,372 links and 13,828 nodes and the transit network consists of 10,298 transit links and 1585 nodes.

In (de Moraes Ramos et al., 2020), a network composed of 520 links and 200 nodes in is considered, using (nested) recursive logit to see how travel information affects route choice behaviour, and what is the impact of the travel time representation on the interpretation of parameter estimates and prediction accuracy.

9.2.2. INVERSE REINFORCEMENT LEARNING ON REAL WORLD TRAVEL TRAJECTORIES

From the field of computer science but similar to Recursive logit, inverse reinforcement learning (IRL) aims to find reward function parameters θ by observing the behaviour of each agent in a Markov decision process (MDP) with a finite set S of N states. The reward function $R(\zeta)$ for trajectory $\zeta = \{s, a\}$, performing action a at state s with f_s the feature vector of state s, is given by:

$$R(\zeta) = \theta^T \mathbf{f}_\zeta = \sum_{s \in \zeta} \theta^T \mathbf{f}_s \tag{9.3}$$

In the computer science literature there are several studies performing IRL on real world problems. (Ziebart et al., 2008) introduced Maximum entropy inverse reinforcement learning in 2008 based on the principle of maximum entropy by (Jaynes, 1957) that the probability of a trajectory ζ with higher reward is exponentially higher than that of a smaller reward: $P(\zeta) \propto e^{R(\zeta)}$. In order to learn from observed behaviour, the maximum entropy IRL algorithm maximizes the likelihood of the observed trajectories under the maximum entropy distribution T

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{\zeta} \log P(\widetilde{\zeta}|\theta, T) \tag{9.4}$$

The maximum entropy distribution T is derived using

$$P(\zeta|\theta) = \frac{1}{Z(\theta)} e^{\sum s_j \in \zeta_i^{\theta^{\mathrm{T}} \mathrm{f}_{s_j}}} \tag{9.5}$$

For parameters θ the partition function $Z(\theta)$ will always converge for the problem with finite horizons and infinite horizon problems with discounted reward weights. Since function 9.4 is convex for a deterministic MDP, gradients for optimizers can be obtained by taking the difference between the observed feature counts and the expected feature counts based on a given set of parameters θ, that can be formulated as the expected state visitation frequencies D_{s_i}. To compute the gradients (Ziebart et al., 2008) uses:

$$\nabla L(\theta) = \widetilde{\mathrm{f}} - \sum_{\zeta} P(\zeta|\theta, T) f_\zeta = \widetilde{\mathrm{f}} - \sum_{s_i} D_{s_i} \mathrm{f}_{s_i} \tag{9.6}$$

To efficiently compute the expected state frequencies for parameters θ, (Ziebart et al., 2008) has proposed an algorithm that approximates the state frequencies by recursively backing up from each possible terminal state, computing each probability mass of each branch along the way, computing partition function Z at each action and state. The branching values give the local action probability that can be used to compute state frequencies and summed up for total frequency counts. (Ziebart et al., 2008) apply the maximum entropy IRL model to learn the reward function of taxi drivers on the road network of Pittsburgh, Pennsylvania. To do so, GPS logging of approximately 7403 trajectories are used to determine the cost of different road type, speed, number of lanes and turn costs. The MDP modeled from the road-network of Pittsburg is assumed to be deterministic with over 300,000 states (street segments) and 900,000 actions (transitions at intersections).

(Hirakawa et al., 2018) use maximum entropy IRL to learn from bird behaviour. As birds are equipped with GPS loggers but gaps may occur due to unavoidable issues with the equipment, a method is needed to fill those gaps with the most likely trajectory. By using maximum entropy IRL to find the reward function they determine the most likely route taken by bids based on environmental features. They applied this approach on one type of bird and found improvement over existing interpolation methods. The IRL model uses 53 trajectories in a 3 dimensional grid world, with 600 cells in height for 200 by 300 grid cells each a square of approximately 3 kilometers wide.

In (Nguyen et al., 2015) a generalization of the IRL problem is proposed that allows multiple locally consistent reward functions to generate the trajectories. By representing the IRL problem with a probabilistic graph model, an expectation-maximization (EM) algorithm can be devised to iteratively learn different reward functions and the stochastic transitions between them, in order to improve the likelihood of the observed trajectories. As a result, the EM algorithm can be used to derive locally consistent reward functions. (Nguyen et al., 2015) empirically evaluated their algorithm with a small real world network and GPS data of 59 taxis in Singapore. In this evaluation the road network is modelled as a simplified grid world with 193 states.

(Mai, Chan, et al., 2019) proposes a generalized version of the causal entropy maximization problem, allowing the possibility to generate a class of maximum entropy IRL models. Their proposed generalized model has the advantage of being able to recover an expert function that would (partially) capture the impact of the connecting structure of the states on experts' decision. Their empirical evaluation on a real-world dataset and a grid-world dataset shows that their generalized model outperforms classical approaches in terms of recovering reward functions and demonstrated trajectories.

(Mai, Nguyen, et al., 2019) proposes a tractable approach to compute directly a log-likelihood of observed trajectories with incomplete/missing data. By performing the training by solving a sequence of linear equations that does not depend on the number of missing segments it is efficient at handling a large number of missing segments. Their empirical evaluation showed that their approach outperforms other approaches.

(Mo, 2019) looks at bicycle route choice applying the maximum entropy IRL approach. To achieve multi-reward functions an extension is used known as Behaviour Clustering IRL (BCIRL). He performs multiple experiments to investigate the applicability of these methods in the context of bicycle route choice. In this study it was found

that a low number of demonstrated trajectories, short trajectory lengths, large number of Markov decision processes to be solved, and class imbalance were problematic issues for the methods. An application was performed on a dataset of GPS trajectories in Amsterdam, but no factors other than distance were found to be relevant.

In 2017 (Wu et al., 2017) proposed a data driven method that construct a MDP that models the decision making process of a public transit rider, decisions such as mode choice, route choice and transfer location choice. The purpose is to predict public transit route choice for urban planners, given various proposed transit construction scenarios. Using this MDP they use maximum entropy IRL to infer the passenger reward function from observed public transit chip card data (AFC) from Shenzhen, China for a period of 3 months. They model the real world as a grid world, dividing the world into grids of a square kilometer and the action set as the possible choice set of different bus and subway routes between each grid cell. The features they include are variables such as fare, travel time, number of transfers and the amount of time remaining to 9 am. In their study they find that they can find a reward function very closed to what is observed with regard to behaviour by public transit users and claim that it justifies their hypothesis that public transit users make sub optimal decisions.

In 2018 (Wu et al., 2018) extended this work in multiple ways to propose a transit evaluation framework. This framework consists of three stages. The first stage data preprocessing divides the urban area into equal size grids, which can be represented as a graph: with the grids as nodes and connected via edges that represent the road and transit system. The second part of the pre-processing consists of aggregating the bus-stops and trajectories into that grid system. The second stage consists of data-driven modelling, modelling the decision process as MDP and derive decision making features from the network such as number of transfers, number of transit options,transit mode, travel time, fare, etc. The final stage is to use this work to learn about rewards, preferences and user choices in order to evaluate transit plans. In the study they describe a preference learning algorithm Inverse Reinforcement Learning with Suboptimal Policy (IRL+SP) that can capture non-linear reward functions of travelers. This algorithm works with the principle of maximum entropy and assumes that experts make decision with soft-max based sub-optimal policies.

To study how well IRL+SP performs in reward learning, (Wu et al., 2018) compare it to IRL and Apprenticeship Learning (AL) and claim that it leads to the lowest ridership vector difference, that IRL and IRL+SP converge faster than AL. To learn how well their algorithm performs in ridership prediction they combine IRL+SP with machine learning techniques such as random forest, lasso regression and linear regression. They compare this with a directly trained machine learning model and to a MNL model and claim to have the lowest prediction relative error. In this study they correctly note that MNL considers a route choice as a single decision of the entire trajectory instead of a sequence of decisions.

9.3. DISCRETE CHOICE MODELLING

In this section we briefly explain the three discrete choice modeling techniques used in this chapter.

9.3.1. MULTINOMIAL LOGIT

In the multinominal logit the utility U for each observation n for each alternative i is

$$U_{ni} = \beta x_{ni} + \varepsilon_{ni} \tag{9.7}$$

$$\varepsilon_{ni} \sim \text{iid extreme value} \tag{9.8}$$

The probability that for observation n alternative i is chosen is:

$$P_{ni} = \frac{e^{\beta_n X_{ni}}}{\sum_j e^{\beta_n X_{nj}}} \tag{9.9}$$

The model is then estimated by maximizing the log-likelihood. First we estimated multinomial logit (MNL) models with a single variable per model and base on those results we removed some variables with high correlation between them. Firstly we decided to include only the highest and lowest level of traffic noise exposure. While the four variables were significant on their own, there was not much difference for the estimated coefficient values between 60 and 70 decibels. Secondly we only included the absolute number of traffic signals as the frequency of traffic signals per kilometer had a lower t-test score. The choice model was estimated both using PandasBiogeme(Bierlaire, 2018).

9.3.2. TASTE VARIATION: MIXED LOGIT

A limitation of the multinominal logit model is that it does not take random taste variation among the thousands of cyclist in our dataset into account. This is why in this study we opted for mixed logit, which is an extension of multi-nominal logit. In the mixed logit the utility is generalized by allowing B_n to be random, this makes the utility of observation n for each alternative i:

$$U_{ni} = \beta_n x_{ni} + \varepsilon_{ni} \tag{9.10}$$

$$\beta_n \sim f(\beta|\theta) \tag{9.11}$$

The probability conditional on β_n that for observation n alternative i is chosen is:

$$L_{ni}(\beta_n) = \frac{e^{\beta_n X_{ni}}}{\sum_j e^{\beta_n X_{nj}}} \tag{9.12}$$

As β_n is random, the choice probability is the integral of the logit formula in equation 9.12 over the probability density function f.

$$P_{ni} = \int L_{ni}(\beta) f(\beta|\theta) d\beta \tag{9.13}$$

9.3.3. PATH SIZE LOGIT

With logit the utility of overlapping paths is overestimated. When δ is large, there is some sort of double counting. The idea of path size logit is to correct for that:

$$V_p = -\beta x_p + \beta \ln PS_p \tag{9.14}$$

where

$$PS_p = \sum_{(i,j)\in p} \frac{c(i,j)}{c_p} \frac{1}{\sum_{q\in C} \delta_{i,j}^q} \tag{9.15}$$

and

$$\delta_{i,j}^q = \begin{cases} 1, & \text{if link (i,j) belongs to path q} \\ 0, & \text{otherwise} \end{cases} \tag{9.16}$$

Where c is the cost function, in study we will simply use length as the cost function.

9.4. CASE STUDY

9.4.1. COLLECTING DATA ON BICYCLE MOVEMENTS

For this study we used the 2016 FietsTelweek ("Bicycle Counting Week") data set (Bikeprint, 2017) that is available at their website. It contains 282,796 unique trips (although the corresponding infographic http://fietstelweek.nl/data/resultaten-fiets-telweek-bekend/ mentions that 416,376 trips having a total distance of 1,786,147 kilometers). During the week of the 19th of September 2016 approximately 29,600 bicyclists volunteered to track their bicycle movements using a smartphone app. For this case study we limited the study to bicycle trips to and/or from the city of Amsterdam, Diemen, Amstelveen and Ouder-Amstel, leaving around 29,684 trips. In Figure 9.1 we visualized all observed trajectories.

This app ran in the background collecting all movements by the bicyclists using the phone's GPS and acceleration sensors. The cyclists used their bike in a way as often seen in the Netherlands, using their bike as transportation from and to work, supermarket, school, etc. For privacy reasons the resulting data was anonymized by the data provider before making it publicly available (i) by the removal of user information to make it impossible to trace multiple trips to a single person and (ii) by rounding of the trip departure time into one-hour bins to the nearest hour and (iii) removal of the random number between 0 and 400 meters from the start and the end of the trip to obfuscate the true origin and destination of each trip.

In prior research based on this data we found in (Koch et al., 2019) that bicyclists in Amsterdam often deviate from the shortest path, more than car drivers, indicating that there are different factors at play in the route choice of bicyclists in Amsterdam. In (Koch et al., 2019) we focused on the concept of route complexity: counting the number of locations where people deviate from the shortest path, in the interest of improving route choice generation techniques and potentially get more insight into the motivations for the route choice for bicyclists.

9.4.2. GENERATION OF ALTERNATIVES

To find out what kind of alternatives exist for each observed path we applied synthetic route choice generation using the Double Stochastic Generation Function (DSGF) method described by (Nielsen, 2000). The DSGF approach produces heterogeneous routes because both the cost and parameters used in the cost function for the links are drawn from a probability function. This way it can generate random paths, just by calculating the shortest path since the cost of each route is based on random factors. (Halldórsdóttir et al., 2014) showed that DSGF has a high coverage level of replicating routes taken by

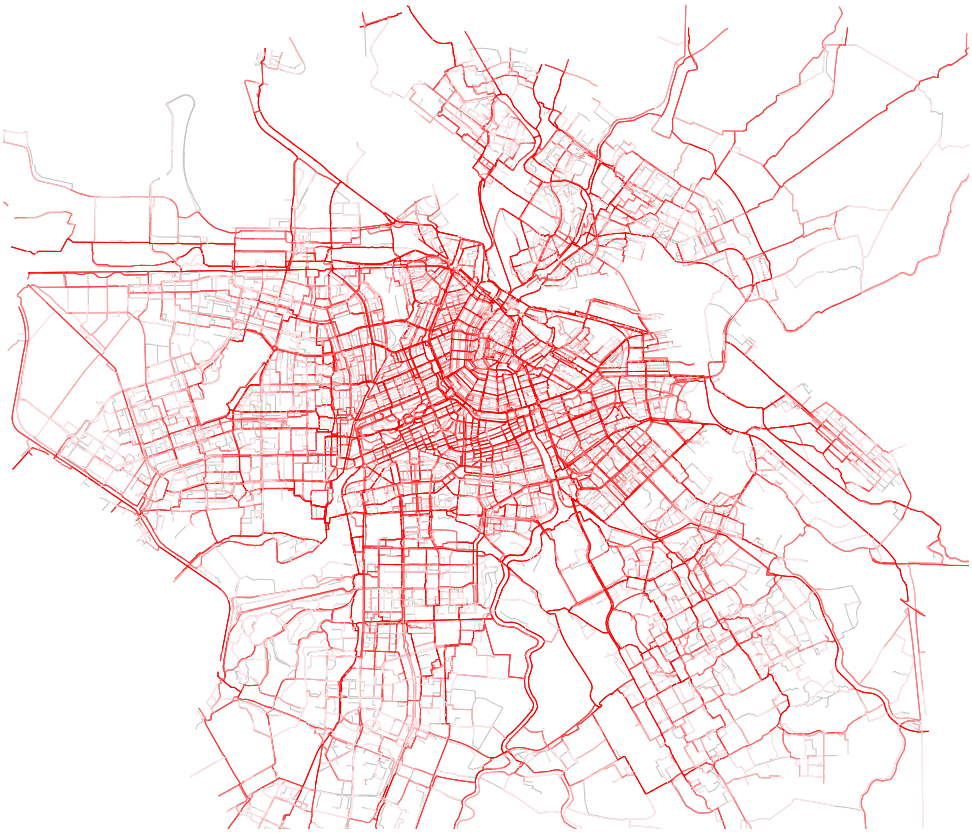


Figure 9.1: Map visualizing the trajectories observed in Amsterdam in the case study.

bicyclists and that it performs well up to 10 kilometer. Furthermore (Bovy & Fiorenzo-Catalano, 2007) state that the method guarantees, with high probability, that attractive routes are included in the choice set, while unattractive routes are left out.

We used an existing implementation of DSGF, specifically POSDAP by (ETH-Zurich, 2012) working on a street network provided by the data collection team of the Fietstelweek, that they imported from OpenStreetMap. We slightly modified POSDAP to execute at most a given number of $M = 128$ iterations (instead of running for a given duration) so that it behaves identically on different machines. For some origin destination pairs POSDAP was not able to find as many as N_0 routes in M iterations, in which case we will use all found routes. The choice sets are written to CSV files for further processing.

Additionally we also run an implementation of Breadth First Search Link Elimination (BFS-LE) by (Rieser-Schüssler et al., 2013), but we opted to leave out these alternatives since there was not much variance in the route choice set. In (Koch et al., 2019) we published on the coverage and consistency of both synthetic choice sets.

Table 9.1: Descriptive statistics on the variables of the observed bicycle trips.

	Min	Median	Avg	Std-dev	Max
Length	500.03	2640.41	4032.32	4935.01	149427.61
55+dB noise	0	0.73	0.68	0.27	1
70+dB noise	0	0.39	0.42	0.29	1
Near green	0	0.18	0.22	0.21	1
Residential	0	0.53	0.52	0.26	1
Near retail	0	0.18	0.22	0.20	1
Near tram	0	0.10	0.24	0.29	1
Tree cover	0	0.40	0.39	0.17	1
Near water	0	0.30	0.34	0.22	1
Traffic signals	0	1	1.99	2.64	25
Cycleway	0	0.53	0.52	0.27	1
$\ln PS$	-2.73	-1.30	-1.28	0.65	0

9.4.3. ENVIRONMENTAL VARIABLES

To collect a set of variables that would reasonably impact route choice of bicyclists we collected and processed open data sources to compute various explanatory variables describing each route. The procedure for the generation of the variables is described below. Descriptive statistics of the variables of the observed trajectories are given in Table 9.1 and for the complete set of generated alternatives in table 9.2.

First of all for each link in the network we include the length of that link as ***distance*** and if that link is a dedicated cycle-way, we include the length as ***oncycleway***. Additionally we have a variable ***traveltime*** based on the length and an estimated speed based on the GPS observations.

To include data about the environment of each link we extracted information of data made openly available by the city of Amsterdam. Firstly we pulled potentially relevant variables from a geographical data-set with land-use zones. To combine the street-network with other relevant geographical data-sets, we cut each street link into small segments of 5 meters and determined the distance of that segments to a geographical feature in the land use data-set. The variable ***nearwater*** measures the distance of street situated close to water bodies such as the canals of Amsterdam, (small) lakes, rivers and other water bodies wider than 6 meters. To determine a preference for routes through parks and forests we did the same thing with the variable ***neargreen***, measuring distance of street situated within a 25 meter radius of 'green' land used for parks, forests and meadows.

For a more fine-grained indication of the level of green and trees along a route, we used a data-set of the location of each individual tree in Amsterdam to determine what portion of each street segment is covered by trees. Our reasoning is that the number of trees has an influence on route choice as they can provide shade on hot days and function as a cover against the wind in storm conditions. To determine the variable ***neartree*** we measured the distance of street within 30 meters left or right from one or more tree(s). This way a street along a row of trees would have the full distance. We determined the

Table 9.2: Descriptive statistics on the variables of the generated alternative bicycle trips.

	Min	Median	Avg	Std-dev	Max
Length	175.59	2663.10	3956.69	4487.86	150758.47
55+dB noise	0	0.74	0.69	0.26	1
70+dB noise	0	0.43	0.43	0.26	1
Near green	0	0.15	0.20	0.19	1
Residential	0	0.54	0.52	0.24	1
Near retail	0	0.18	0.23	0.19	1
Near tram	0	0.15	0.24	0.26	1
Tree cover	0	0.39	0.39	0.16	1
Near water	0	0.27	0.31	0.22	1
Traffic signals	0	1	2.19	2.72	28
Cycleway	0	0.48	0.47	0.24	1
$\ln PS$	-2.78	-1.45	-1.44	0.49	0

distance of 30 meters between road and tree based on various situations where a rows of trees are situated along bicycle roads in Amsterdam.

To measure the effect of residential areas the variable ***nearresidential*** measures the distance of streets in residential areas. . The variable ***nearretail*** describes distance within areas purposed for *'Shops, malls and hotels-restaurants-pubs'*, *'Public offices and services'* and *'Cultural, social, medical, educational'*.

To see if the vicinity of busy roads, a major source of noise and pollution, has any impact on route choice we used a data set with the noise contours map of road traffic in Amsterdam as shown in Figure 9.2. This data-set is produced by a model that estimates the level of exposure to traffic noise in this map there are four noise levels with respectively at least 55, 60, 65 or 70 decibels of noise. The variables ***nearXdb*** represent the distance of the street passing through these exposure zones.

Based on the idea that tramlines in Amsterdam form a radial artery towards the heart of the city, we construct the variable ***neartram*** indicating the portion of the route that is situated 100 meter from tram rails either to the left or right of the path, measured using segments of 10 meters.

Finally we wanted to see if the number and frequency of traffic signals has a measurable effect on route choice. We included this in two ways: first the exact number of traffic signals with ***ntrafficsignal*** and secondly the frequency of traffic signals ***trafficsignalfreq*** where the number of signals is divided by the length of the route.

Since Amsterdam has no elevation changes beyond the occasional bridge, we did not include any elevation changes as a variable.

9.5. RESULTS

MULTINOMIAL LOGIT

The estimated parameters for our baseline multinominal logit model are presented in the first column of 9.3. The results are mostly as expected except for length. A Traffic signals and being near roads with (heavy) noise emission all have a negative utility. Being

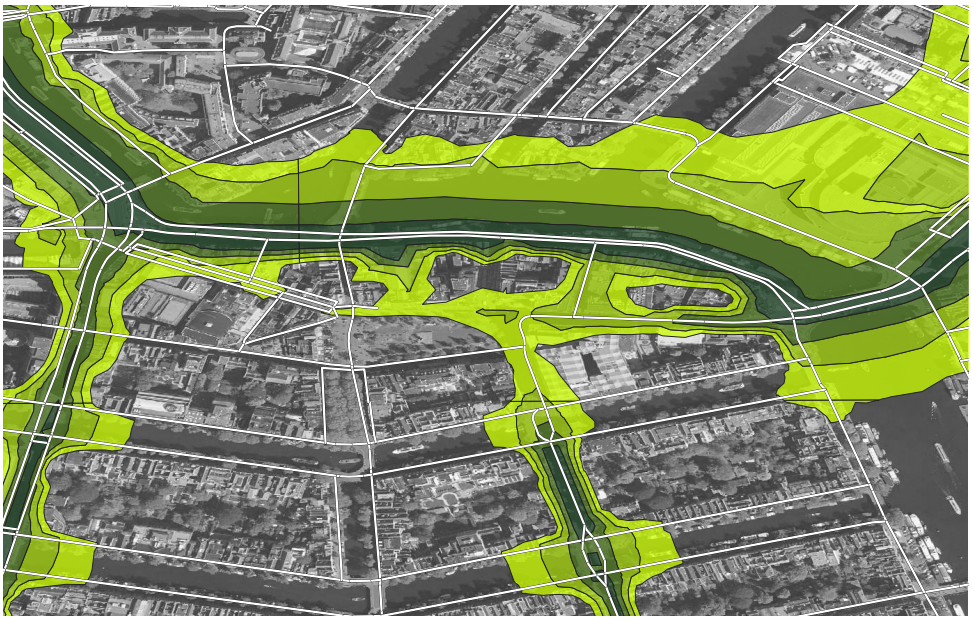


Figure 9.2: Noise contour map of Amsterdam, used for the variable that indicates the distance of a trajectory along roads with noisy traffic.

close to water and/or green land-use and dedicated cycle-way infrastructure all have a strong positive utility. Being near retail land-use has a positive utility while residential land-use has a negative utility. The positive utility of being near tram lines may be a correlation effect of tramlines in Amsterdam forming a guide way into the city center. The unexpected positive effect of additional distance warrants further investigation by adding mixing for taste variation.

MIXED LOGIT

The results for the series of mixed logit models are listed in the second through twelfth columns of Table 9.3. These results are based on running the model with 5000 draws. In each column we list the estimated parameters for a mixed logit model with the standard deviation (sigma σ) on the variable indicated in the header of the column. The estimated standard deviation (sigma) for that variable is listed at the bottom of the table, this is not applicable for the baseline multinomial logit. model. The log likelihood for each model is listed in Table 9.4.

PATH SIZE LOGIT

In Table 9.5 we listed results for models we ran with the path size logit as a variable. The results for a multi-nominal logit and mixed logit models were estimated using 1000 draws. The log likelihoods are listed in Table 9.6. We also see that the natural log of path size variable is a statistically significant in every model that we estimated. Again just like as in the mixed logit we see here that the model with taste variation on route length has the best log likelihood.

9.6. RECURSIVE LOGIT MODEL EXPERIMENTS

In this section we describe a series of experiments in modeling bicycle route choice in the city of Amsterdam using recursive logit model with the observed bicycle routes in our data, environmental explanatory variables, and the Amsterdam street network – without generating route choice alternatives.

9.6.1. RECURSIVE LOGIT WITH ENVIRONMENTAL VARIABLES

Our initial attempt was to model the Amsterdam network with each intersection as a node and the streets as actions, following example in (Zimmermann & Frejinger, 2020). This resulted in a network with approximately 46,000 links and 30,000 observations, which we carefully controlled for full connectivity and no isolated graphs. Our motivation to model intersections as states instead of links as states was driven to lower the number of total states to be modeled, under the assumption that turn angles might have a low influence on bicycle route choice in Amsterdam. We tested the recursive logit model with the five variables *length*, *oncycleway*, *nearwater*, *neargreen* and *near55db*. However we were unable to get the solver to give plausible results for equation 9.1 as the solver would return incorrect results.

Table 9.3: Baseline multinomial logit and series of mixed logit models with taste variation per explanatory variable.

	MNL	**Route Length**	**55+dB Noise**	**70+dB Noise**	**Near green**	**Residential**	**Near retail**	**Near tram**	**Tree cover**	**Near water**	**Traffic signals**	**Cycle way**
βroute length	1.86	-2,21	1.67	1,66	1,76	1,74	1,77	1,61	1,76	1,77	1,8	1.9
t-test	62.1	-14,4	51.6	50,7	55,2	54.5	57,2	50,8	54,3	53,9	57,2	56.2
β55+dB noise	-0.324	-0,496	0.484	-0,711	-0,284	-0,0492	-0,32	-0,403	-0,046	-0,519	-0,348	-0.356
t-test	-3.32	-3,91	3.36	-6,06	-2,7	-0,45	-3,05	-3,66	-0,416	-4,57	-3,47	-3.15
β70+dB noise	-2.09	-2,54	-2.68	-2,25	-2,28	-2,37	-2,13	-2,42	-2,39	-2,08	-2,2	-2.52
t-test	-26.3	-24,9	-29.2	-18,1	-26,4	-26,8	-25,1	-26,3	-26,7	-22,6	-26,8	-26.9
βnear green	2.47	2,84	2.73	3,1	3,73	2,79	2,59	2,69	2,68	2,59	2,43	2.85
t-test	28.9	25,3	27.5	30,1	29	28,7	28,7	27,9	27,4	26,2	27,8	29.1
βresidential	-0.373	-0,415	-0.094	-0,263	-0,261	-0,259	-0,43	-0,0838	-0,224	-0,511	-0,376	-0.324
t-test	-5.51	-4,83	-1.2	-3,3	-3,54	-2,62	-5,86	-1,1	-2,96	-6,59	-5,39	-4.18
βnear retail	0.639	0,921	0.967	0,994	0,592	1,14	0,612	0,56	0,818	0,776	0,589	0.933
t-test	6.9	7,95	9.24	9,26	6,02	10,8	4,79	5,39	8	7,35	6,15	8.96
βnear tram	1.3	1,89	1.34	1,37	1,59	1,5	1,41	0,655	1,59	1,45	1,38	1.67
t-test	19.2	21,7	17	16,6	21,3	19,9	19,4	5,43	20,9	18,5	19,5	21.2
βtree cover	1.46	2,44	2.17	1,56	1,65	1,71	1,66	1,77	3	2,27	1,71	1.81
t-test	11.5	14,7	14.3	10	11,7	11,9	12,3	12,1	15,5	15,4	13	12.5
βnear water	2.11	2,22	2.26	1,93	2,16	2,31	2,23	2,15	2,29	2,11	2,09	2.09
t-test	36	29,5	33.6	27,7	33,7	35,1	35,5	33	34,8	21,2	34,6	30.8
βtraffic signals	-0.496	-0,501	-0.591	-0,733	-0,465	-0,499	-0,513	-0,649	-0,501	-0,479	-1,21	-0.476
t-test	-9.84	-7,61	-10.6	-12,5	-8,86	-9,28	-9,75	-11	-9,33	-8,71	-15,4	-8.64
βcycleway	4.3	4,63	4.91	4,87	4,61	4,67	4,55	4,89	4,71	4,86	4,35	6.43
t-test	78	70	78.5	76,4	77,8	77	77,7	79	77,5	78,3	76,9	64.8
σ	N/A	16,9	10.6	10,3	8,93	8,22	8,94	10,1	15,8	9,3	4,89	6.53
t-test	N/A	65,9	50.2	57,4	38,8	44,6	34,7	50,4	46,6	52,7	35	64.7

Table 9.4: Log likelihood of baseline multinomial logit and series of mixed logit models with taste variation per explanatory variable.

	MNL	**Route Length**	**55+dB Noise**	**70+dB Noise**	**Near green**	**Residential**	**Near retail**	**Near tram**	**Tree cover**	**Near water**	**Traffic signals**	**Cycle way**
LL	-79817	-75922	-78500	-77851	-79215	-79028	-79407	-78092	-78963	-78478	-79365	-77705

Table 9.5: Baseline multinomial logit and series of mixed logit models with path size and taste variation per explanatory variable.

	MNL	**Route Length**	**55+dB Noise**	**70+dB Noise**	**Near green**	**Resid-ential**	**Near retail**	**Near tram**	**Tree cover**	**Near water**	**Traffic signals**	**Cycle way**
βroute length	1.27	-5.03	1.33	1.37	1.29	1.29	1.29	1.29	1.3	1.37	1.28	1.44
t-test	38.4	17.8	36.5	36.7	36.7	36.6	37.7	36.3	36.4	37	36.8	37.7
β55+dB noise	-0.32	-0.353	0.41	-0.681	-0.291	-0.081	-0.312	-0.394	-0.075	-0.504	-0.337	0.334
t-test	-3.39	-2.85	2.98	-5.93	-2.87	-0.766	-3.1	-3.66	-0.702	-4.56	-3.48	-3.03
β70+dB noise	-1.93	-2.3	-2.54	-2.17	-2.13	-2.23	-2	-2.3	-2.25	-1.97	-2.04	-2.37
t-test	-25	-23.1	-28.4	-18.1	-25.5	-25.9	-24.4	-25.5	-25.9	-21.9	-25.6	-25.9
βnear green	2.35	2.76	2.66	3.03	3.47	2.69	2.47	2.61	2.58	2.51	2.33	2.76
t-test	28.5	25.2	27.5	29.9	28.7	28.7	28.3	27.7	27.4	26.1	27.5	28.9
βresidential	-0.347	-0.395	-0.103	-0.264	-0.263	-0.262	-0.408	-0.086	-0.225	-0.493	-0.351	-0.317
t-test	-5.3	-4.7	-1.35	-3.38	-3.7	-2.81	-5.78	-1.15	-3.06	-6.54	-5.18	-4.18
βnear retail	0.665	0.953	-0.964	0.996	0.632	1.11	0.631	0.579	0.826	0.775	0.606	0.931
t-test	7.41	8.42	9.42	9.46	6.64	10.8	5.34	5.69	8.32	7.53	6.53	9.15
βnear tram	1.31	1.97	1.36	1.38	1.55	1.49	1.4	0.716	1.56	1.45	1.37	1.67
t-test	20.1	23.2	17.8	17.1	21.6	20.5	20.1	6.22	21.2	19	20.1	21.6
βtree cover	1.42	2.57	2.1	1.53	1.59	1.64	1.59	1.73	2.78	2.19	1.63	1.77
t-test	11.6	23.2	17.8	10	11.7	11.8	12.1	12.1	15.1	15.2	12.8	12.4
βnear water	2.03	2.13	2.2	1.9	2.09	2.23	2.16	2.11	2.21	2.05	2.02	2.04
t-test	35.6	28.8	33.6	27.8	33.7	35	35.4	33.1	34.6	21.7	34.5	30.8
βtraffic signals	-0.477	-0.439	-0.575	-0.712	-0.449	-0.485	-0.494	-0.635	-0.486	-0.468	-1.13	-0.466
t-test	-9.68	-6.72	-10.5	-12.3	-8.76	-9.22	-9.62	-10.9	-9.25	-8.67	-15.1	-8.6
βcycleway	4.26	4.6	4.86	4.85	4.55	4.63	4.48	4.86	4.66	4.82	4.31	6.32
t-test	78.6	70.2	78.8	76.8	78.2	77.6	78.1	79.3	78	78.6	77.6	65.2
βln PS	0.56	1.06	0.365	0.313	0.473	0.458	0.484	0.339	0.462	0.396	0.496	0.439
t-test	30.5	37.9	17.2	14.3	23.9	22.6	24.6	16.1	22.8	18.8	26	20.9
σ	N/A	17.8	9.71	9.7	7.77	7.32	7.45	9.35	14.3	8.58	4.31	9.08
t-test	N/A	67.3	45.7	58.3	34	39.7	28.5	47.3	42.1	48.7	30.8	62.1

Table 9.6: Log Likelihood multinomial logit and series of mixed logit models with path size and taste variation per explanatory variable.

	MNL	**Route Length**	**55+dB Noise**	**70+dB Noise**	**Near green**	**Resid-ential**	**Near retail**	**Near tram**	**Tree cover**	**Near water**	**Traffic signals**	**Cycle way**
LL	-79372	-75221	-78356	-77758	-78951	-78793	-79123	-78030	-78805	-78307	-79044	-77501

9.6.2. PYTHON RE-IMPLEMENTATION OF RECURSIVE LOGIT

For the purpose of a better understanding of the algorithm we also implemented our own version of the original recursive logit model in (Fosgerau et al., 2013) and the significantly faster decomposition recursive logit model (Mai et al., 2018) in Python with SciPy and NumPy. We were indeed able to successfully replicate models with data available online using our Python re-implementation. However also our Python re-implementation was not effective in giving plausible estimation results for the Amsterdam bicycle case.

9.6.3. REDUCTION OF STUDY AREA TO AMSTERDAM CITY CENTER ONLY

Subsequently we simplified the study area to just the Amsterdam city center area containing only about 4500 links, excluding the entire municipality and surrounding suburbs. Again we carefully controlled for full connectivity and no isolated graphs. This too did not lead to plausible estimation results. We visualized this specific selection in Figure 9.3.

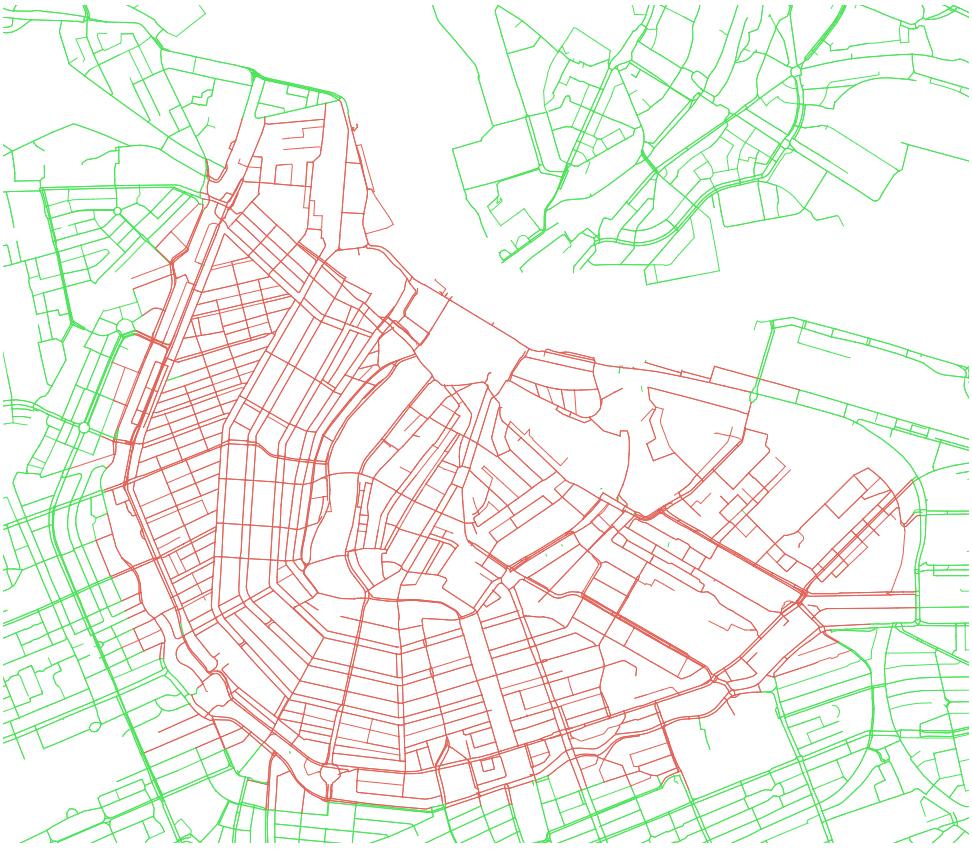


Figure 9.3: Map showing the selected area for city center of Amsterdam in red.

9.6.4. REDUCTION OF STREET NETWORK COMPLEXITY

Based on the remark by (Fosgerau et al., 2013) on dense networks and the number of alternative paths, our next action was to simplify the street network in the Amsterdam city center and remove all footpaths to reduce the complexity of the network. Again we carefully controlled for full connectivity and no isolated graphs. We accordingly also removed all observations of GPS trajectories cycling over footpaths. This too did not lead to plausible estimation results.

9.6.5. EDGE-BASED NETWORK VERSUS INTERSECTION-BASED NETWORK

Finally to transform our model to a model more similar to the studies in the literature, we instead created an edge-based network, instead of the intersection-based network. In the adapted implementation, each state is a street-segment and each action is a move to another street segment. This link-link approach allows the possibility to create new features with a boolean to indicate turns, left turns and u-turns, similar to the Borlänge model in (Fosgerau et al., 2013) and (Mai et al., 2018). For the entire city of Amsterdam

Table 9.7: Results from models. that were estimated on 987 observations in the city center of Amsterdam.

Utility parameter	Estimated value	t-Test
Travel time (minutes)	-2.9119	-19.7751
Intersections	-0.6356	-13.8377
Left turn	-1.5717	-20.8542
U-turns	0.4205	7.5003
Log likelihood	-4.489078	
Travel time (minutes)	-18.0368	-2587.03
Log likelihood	-10.279184	
Travel time (minutes)	-18.03681	-2587.035
Distance exposed to traffic noise >= 55db	-1.9330	-0.1354
Log likelihood	-10.279184	
Travel time (minutes)	-18.0369	-2580.7
Distance along water (km)	-2566.333	-3.7053
Log likelihood	-10.274874	
Travel time (minutes)	-18.0368	-2587.0
Distance along trees (km)	-1.9330	-0.0098
Log likelihood	-10.279184	

this model contains 40063 links as states with 137724 transitions between states; for the city center area only it consists of 4204 links as states with 15234 transitions.

EDGE-BASED NETWORK ON ENTIRE CITY OF AMSTERDAM: TURN VARIABLES

With this network we were now able to solve the linear system to obtain a solution of z without (invalid) negative values for the entire city of Amsterdam. However even when setting the maximum number of links per observation at 30 links, we are still unable to calculate a log likelihood due to values of $z_{origin} == 0$ for one or more of the observations.

EDGE-BASED NETWORK ON AMSTERDAM CITY CENTER ONLY: TURN VARIABLES

In the smaller area of the city center of Amsterdam it is possible to estimate a model but also with a relatively low limit of 30 links per observation, as a higher limit would again return zero values for z_{origin}. This meant we are able to process only 987 observations and 681 destinations. We listed the results of this model in Table 9.7 where we would describe the betas to be plausible. An increase travel time would be a obvious cost. The negative value of an intersection, especially in the city center where almost all road/bicycle intersections are equipped with traffic lights is also expected. The positive reward for for left-turn seems as expected to avoid crossing traffic. The positive reward for u-turn may seem odd, but u-turn costs on a bicycle should be less costly than in a motorized vehicle.

EDGE-BASED NETWORK ON AMSTERDAM CITY CENTER: ENVIRONMENTAL VARIABLES

Subsequently we modelled travel-time with our five variables separately: 70db traffic noise distance, near green distance, near water distance, tree covered distance, cycle way distance. While all these models did converge, it was only able to invert the hessian to calculate a standard error for models: travel-time and length with tree cover; travel-time and length along water; travel-time and length with traffic noise. We listed these results in Table 9.7. The βlength – tree cover for is not significant, possibly because tree cover is less of an issue in the city center which is shielded by buildings. The result for traffic noise is not significant either. The only significant effect found besides travel-time was the distance travelled near water, possibly due to cycling along the cobble-stone paved narrow canals navigating between cars, trucks and tourists being perceived as disadvantageous to persons who cycle for daily activities.

9.7. DISCUSSION

Given our experience with the Amsterdam model, we highlight several challenges during the estimation of the recursive logit model. We reflect on why our initial plan for estimating the bicycle choice behavior in the entire city of Amsterdam with environmental variables was feasible with the baseline multinomial logit model, but faced numerical complications with the recursive logit model.

9.7.1. NEGATIVE REWARD FORMULATION

In the original paper by (Fosgerau et al., 2013) on recursive logit it is mentioned that to formulate the path choice problem as a dynamic discrete choice model with the utility maximization problem consistent with a dynamic programming problem, the deterministic utility component is required to have negative value: $v_n(a|k) = v(x_{n,a|k;\beta}) < 0$.

As an experiment we set up a network based on a simple grid layout, with 625 intersections, allowing the user to move left, right, up, down. There is one diagonal connection across from the top left corner to the bottom right corner. We included each segment between the intersections as a single unit of distance. See Figure 9.4 for a visualization of 10 by 10 grid. We set up 4 different variables: $\beta_{distance}$ for the unit distance, $\beta_{intersection}$ that counts each intersection passed, β_{left} that counts each move towards the left side of the grid, $\beta_{diagonal}$ that counts each diagonal move. We included two observations across the top and right of the grid and a observation across the left and bottom of the grid and a series of 10 random observations that have a strong preference to move diagonally when possible. This model estimated with a log likelihood of -10 and $\beta_{distance} = -1.54467129, \beta_{intersection} = -2.04467129, \beta_{diagonal} = -2.09161539, \beta_{left} = -81.34025$.

What we observed is that altering the attribute of a single link of this model to make the utility of that link positive lead to the inability of the linear solver to return a valid solution and thus not being able to find a log likelihood or estimate a model.

An implication that when using recursive logit you should aim for only including costs in your function u. In practice this might turn out tricky as cost variables may turn out to be correlated with reward variables not included in your model. For example heavy traffic near a bicycle path may seem like a cost variable at first, but as such roads

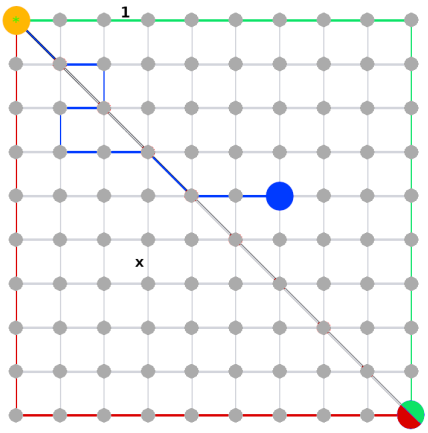


Figure 9.4: Two fixed paths to same destination along the boundaries of the graph (in red and green), plus example of one randomly generated path (in blue). All paths start at the top left corner and end respectively at the large red/green circle and the large blue circle.

are likely equipped with street lights in contrast to a path through a dark and empty park, such variable may turn out to have a negative cost.

9.7.2. VALID INITIAL PARAMETERS AND LENGTH OF OBSERVATIONS

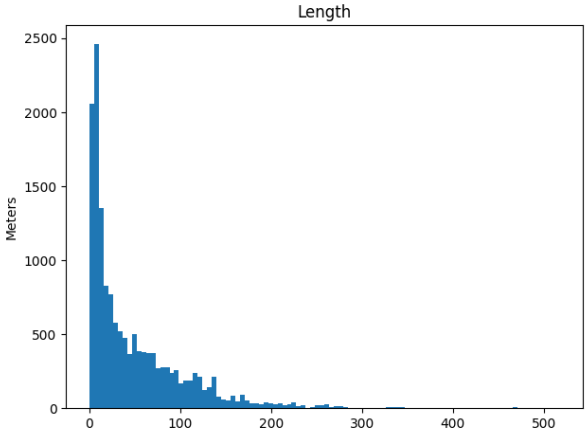
To take a closer look at how difficult it can be to determine a valid initial parameter prior to iterative solution of the system, we proceeded to look at solely at travel-time without any other features in the model. To do so, we manually computed the log-likelihood function for a range of the $\beta_{travel-time}$ parameter in the range between -1 and -25. We saw that only in a small window of $\beta_{travel-time}$ between approximately -18.02 and -21.01 a valid log likelihood function exists. For a $\beta_{travel-time} <= -18$ the equation system would return an invalid sign for the log likelihood, for $\beta_{travel-time} >= 21.01$ at least one of the observations would return a $exp(V) = 0$ for at the starting value.

This narrow range was achieved with a number of links in each observations limited at 40. If we allowed observations with more links we were unable to find a window of initial parameters where the log likelihood function is valid at all. We see numerical issues as the root cause of this. As a long recursion will be a sum of each link utility, with high values due to the exponential, we expect these results to be caused by overflows and under flows in the solver.

9.7.3. THE DISTRIBUTION OF VALUES OF FEATURES AND NETWORK DEGREE CENTRALITY

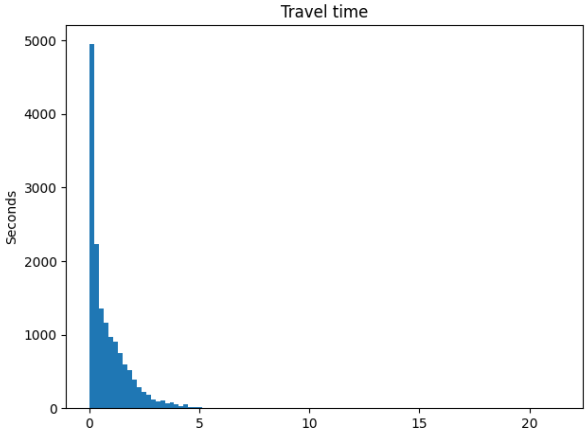
Subsequently, we attempted a similar experiment with the only feature in the model being β_{length}, which is correlated with $\beta_{travel-time}$. We were unable to find an exact parameter of β_{length} that is valid, but deduce it is somewhere between -413.6 and -413.7, based on where the solver returns a valid solution but $exp(V) = 0$. To look at the differ-

ence between both variables we refer to descriptive statistics presented earlier in Table 9.1 and histograms of the distributions plotted in Figure 9.7. Based on the descriptive



8cm

Figure 9.5: Length



8cm

Figure 9.6: Travel time

Figure 9.7: Histogram of the variables length and travel time of the bicycle network in the city center of Amsterdam.

statistics we expect the same root cause that makes it difficult to find a valid starting parameter. The lower kurtosis in the distribution of the length indicates a fatter right-tailed distribution presenting more possibility for a significant number of relatively large values to end up added together in the recursion on links. This too can lead to overflows and under flows making it difficult to find starting values betas due to numerical issues.

9.7.4. THE NUMBER OF ALTERNATIVE CHOICE OPTIONS

Another difference with existing studies in the literature that due to the complexity of bicycle infrastructure in Amsterdam, the number of possible options is higher than we would see in car route choice or in a city without two cycle-paths on both sides of major roads or two roads in both directions (for cyclists) along the canals.

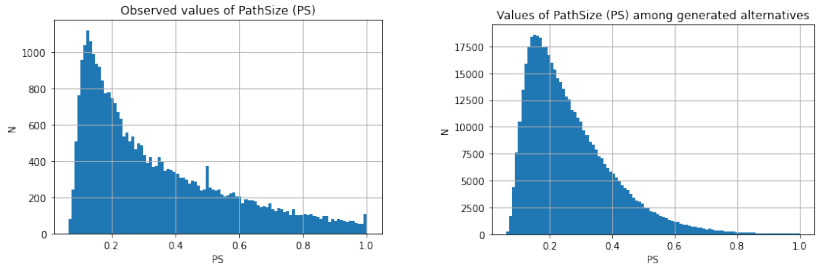
9.7.5. DISCRETE CHOICE MODELS

Based on the log likelihood ratio-tests conducted and included in Table 9.8 we see that each extension brings a improvement of the log-likelihood at a very high level of statistical significance.The series of mixed logit models with path size correction are the best performing model.

We also see that adding the taste variation on the length attribute results in the largest improvement, this intuitively makes sense as large number of anonymous cyclists observed all have their own taste for how much distance they are prepared to cycle. This leads to different tolerances to base behavior on attributes such as land-use and cycleway infrastructure.

A limitation in this study is that the cyclists data used in this study was anonymous, so we were not able to group multiple observations with the same taste preferences of an individual using with a panel mixed logit model. A panel mixed logit model likely will likely yield a further improvement of the model.

While we see statistical significant improvement of path-size, we also see that it brings a smaller improvement than taste variation. This is likely due to a low overlap in the alternatives generated by the route choice generator that we used, namely POSDAP(ETH-Zurich, 2012) with doubly stochastic choice set generation. This is also shown in the distributions of path sizes in observed and generated paths as shown in Figure 9.8a and 9.8b. The low overlap is what we are already found in our earlier study on the quality of these choice sets in in our previous work in (Koch et al., 2019).



(a) Observed trajectories (b) Generated alternatives

Figure 9.8: Histograms of PathSize (PS) on observed trajectories and in the collection of generated alternatives

Our cycling models show that cyclists route choice comes with a high level of stochasticity: how much distance bicyclists are prepare to take detours can widely vary person by person. This can make it hard to simulate route choice by bicyclists, harder than the route choice for car drivers, as shown in our previous study (Koch et al., 2019) on route

complexity for bicyclists and car drivers.

9.8. CONCLUSION

Recursive logit is a promising solution for inverse reinforcement learning on specific route choice problems. However when designing your model and variables it is very important to keep the limitations of the linear equation system in mind. These limitations can make it impossible to estimate your model or lead to wrong estimations.

As recursive logit may fail to converge if even a single link has a (high) reward instead of cost, it is important to think through whether your variables are always costs for all links in the network. This can be hard in practice, as assumptions can be deceiving. For example you might model a bridge as a cost, as there is a small slope involved, however in reality people might prefer a route over a bridge as a form of sight seeing opportunity. Furthermore preferences can differ by person or vary over the time of day. For example a park might be a beneficial detour during the day, but during the night an empty badly lit park that feels unsafe might be worth a detour around instead.

DISCRETE CHOICE MODELLING

In this chapter we have estimated a route choice model on 29,684 bicyclist trips conducted to/from the City of Amsterdam, Diemen, Amstelveen and Ouder-Amstel. We have generated alternatives routes using the doubly stochastic generation function in POSDAP (ETH-Zurich, 2012). Using this choiceset we have estimated 3 kind of discrete choice models, a simple multinomial logit model, mixed logit and both extended with the path size logit variable. Our results show that the mixed logit with path size correction and taste variation on length results in the best performing model.

This model shows that bicyclists are prepared to make detours to use dedicated cycleway infrastructure, avoid roads near heavy noise (70 decibels or more) emitting roads and avoid traffic signals. Cyclists prefer roads along tram lines, probably a correlation with how cyclists find their way into the city center. Cyclists prefer to ride along water, under the cover of trees and along green land-use zones such as parks. An interesting finding is that cyclists prefer to cycling across retail land-use while avoiding residential land use.

In a future study we would like to repeat our work on a data-set where we can identify individuals in order to estimate a panel mixed path-size logit model.

9.9. FUTURE STUDY

For future study we are interested in the precise computational details that lead to the invalid estimates by the solver when faced with numerical overflow and underflow issues.

We are also looking into how well extensions of algorithms based on maximum entropy IRL of (Ziebart et al., 2008) will function with the Amsterdam bicycle network given the successful implementation of inverse reinforcement learning for bicycle paths in the work by (Mo, 2019), however as in this study similar limitations were noted regarding the size of the state space.

In our initial experiments with implementing maximum entropy IRL by (Ziebart et

Table 9.8: Overview of log likelihood ratio tests

Reference Group(s)	d.f.	LR	LU	-2[LR- LU]	χ^2 d.f.(0.05)	p-value	Comments
LR Test: Mixed logit (unrestricted) vs. MNL (restricted)							
Length	1	-79817	-75922	7790	3,84	0	Reject restrictions
55+ dB noise	1	-79817	-78500	2634	3,84	0	Reject restrictions
70+ dB noise	1	-79817	-77851	3932	3,84	0	Reject restrictions
Near green	1	-79817	-79215	1204	3,84	0	Reject restrictions
Residential	1	-79817	-79028	1578	3,84	0	Reject restrictions
Near retail	1	-79817	-79407	820	3,84	0	Reject restrictions
Near tram	1	-79817	-78092	3450	3,84	0	Reject restrictions
Tree cover	1	-79817	-78963	1708	3,84	0	Reject restrictions
Near water	1	-79817	-78478	2678	3,84	0	Reject restrictions
Traffic signal	1	-79817	-79365	904	3,84	0	Reject restrictions
Cycleway	1	-79817	-77705	4224	3,84	0	Reject restrictions
LR Test: Mixed logit with Pathsize (unrestricted) vs. MNL with Pathsize (restricted)							
Length	1	-79372	-75221	8302	3,84	0	Reject restrictions
55+ dB noise	1	-79372	-78356	2032	3,84	0	Reject restrictions
70+ dB noise	1	-79372	-77758	3228	3,84	0	Reject restrictions
Near green	1	-79372	-78951	842	3,84	0	Reject restrictions
Residential	1	-79372	-78793	1158	3,84	0	Reject restrictions
Near retail	1	-79372	-79123	498	3,84	0	Reject restrictions
Near tram	1	-79372	-78030	2684	3,84	0	Reject restrictions
Tree cover	1	-79372	-78805	1134	3,84	0	Reject restrictions
Near water	1	-79372	-78307	2130	3,84	0	Reject restrictions
Traffic signal	1	-79372	-79044	656	3,84	0	Reject restrictions
Cycleway	1	-79372	-77501	3742	3,84	0	Reject restrictions
LR Test: MNL with Pathsize (unrestricted) vs. MNL (restricted)							
Baseline	1	-79817	-79372	890	3,84	0	Reject restrictions
LR Test: Mixed logit with Pathsize (unrestricted) vs. Mixed logit (restricted)							
Length	1	-75922	-75221	1402	3,84	0	Reject restrictions
55+ dB noise	1	-78500	-78356	288	3,84	0	Reject restrictions
70+ dB noise	1	-77851	-77758	186	3,84	0	Reject restrictions
Near green	1	-79215	-78951	528	3,84	0	Reject restrictions
Residential	1	-79028	-78793	470	3,84	0	Reject restrictions
Near retail	1	-79407	-79123	568	3,84	0	Reject restrictions
Near tram	1	-78092	-78030	124	3,84	0	Reject restrictions
Tree cover	1	-78963	-78805	316	3,84	0	Reject restrictions
Near water	1	-78478	-78307	342	3,84	0	Reject restrictions
Traffic signal	1	-79365	-79044	642	3,84	0	Reject restrictions
Cycleway	1	-77705	-77501	408	3,84	0	Reject restrictions

al., 2008) on even the simplified problem of the Amsterdam city center with observations with a length of less than 30 links, we encountered overflows when calculating the local action probabilities as the expected reward for each state grew exponentially even when applying a discount factor.

Our recommendation would consist of searching for a way to dramatically simplify the state space by find ways to abstract the decision process. One possibility could be for example applying principles of path complexity such as in (Koch et al., 2019).

BIBLIOGRAPHY

Agrawal, D., & Schorling, C. (1996). Market share forecasting: An empirical comparison of artificial neural networks and multinomial logit model. *Journal of Retailing*, *72*(4), 383–408.

Anderson, P., Owen, A., & Levinson, D. (2013). The time between: Continuously-defined accessibility functions for scheduled transportation systems. *Proceedings of the 92nd Annual Meeting of the Transportation Research Board*.

Arnott, R., & Inci, E. (2006). An integrated model of downtown parking and traffic congestion. *Journal of Urban Economics*, *60*(3), 418–442.

Arthur, S. (1955). Parking structure device [US Patent 2,727,638].

Badii, C., Nesi, P., & Paoli, I. (2018). Predicting available parking slots on critical and regular services by exploiting a range of open data. *IEEE Access*, *6*, 44059–44071.

Barak, S., Nasiri, M., & Rostamzadeh, M. (2019). Time series model selection with a meta-learning approach; evidence from a pool of forecasting algorithms. *arXiv preprint arXiv:1908.08489*.

Barceló, J., Gilliéron, F., Linares, M., Serch, O., & Montero, L. (2012). Exploring link covering and node covering formulations of detection layout problem. *Transportation Research Record: Journal of the Transportation Research Board*, (2308), 17–26.

Barth, M., & Boriboonsomsin, K. (2007). Real-world co2 impacts of traffic congestion. *Proceedings of the 87th Annual Meeting of the Transportation Research Board*.

Ben-Akiva, M., & Bierlaire, M. (1999). Discrete choice methods and their applications to short term travel decisions. *Handbook of transportation science* (pp. 5–33). Springer.

Benenson, I., Martens, K., Rofé, Y., & Kwartler, A. (2011). Public transport versus private car gis-based estimation of accessibility applied to the tel aviv metropolitan area. *The Annals of Regional Science*, *47*(3), 499–515.

Bentz, Y., & Merunka, D. (2000). Neural networks and the multinomial logit for brand choice modelling: A hybrid approach. *Journal of Forecasting*, *19*(3), 177–200.

Bernardi, S., La Paix Puello, L., & Geurs, K. (2018). Modelling route choice of dutch cyclists using smartphone data. *Journal of transport and land use*, *11*(1), 883–900.

Bierlaire, M. (2018). *Pandasbiogeme: A short introduction* (tech. rep.). Technical Report TRANSP-OR 181219, Transport and Mobility Laboratory, EPFL.

Bikeprint. (2017). Download bestanden Nationale Fietstelweek 2015 en 2016. http://www.bikeprint.nl/fietstelweek/

Blanchard, S. D., & Waddell, P. (2017). Urbanaccess: Generalized methodology for measuring regional accessibility with an integrated pedestrian and transit network. *Transportation Research Record*, *2653*(1), 35–44.

Bovy, P. H., & Fiorenzo-Catalano, S. (2007). Stochastic route choice set generation: Behavioral and probabilistic foundations. *Transportmetrica*, *3*(3), 173–189.

Bovy, P. H., & Hoogendoorn-Lanser, S. (2005). Modelling route choice behaviour in multi-modal transport networks. *Transportation*, *32*(4), 341–368.

Bradley, M. A., & Gunn, H. F. (1990). Stated preference analysis of values of travel time in the netherlands. *Transportation Research Record*, (1285), 78–88.

Broach, J., Dill, J., & Gliebe, J. (2012). Where do cyclists ride? a route choice model developed with revealed preference gps data. *Transportation Research Part A: Policy and Practice*, *46*(10), 1730–1740.

Brownlee, J. (2019). A gentle introduction to SARIMA for time series forecasting in Python. https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/

Buijs, R., Koch, T., & Dugundji, E. (2020). Using neural nets to predict transportation mode choice: An Amsterdam case study. *Procedia Computer Science*, *170*, 115–122.

Bundesbahnen, S. (2020). GitHub matsim-extensions by sbb [Accessed: 2019-12-01].

Byrd, A., & Emory, D. (2012). K. webb. open trip planner analyst: Open source, open data driven transportation network analysis. poster p13-6039, session 616 (open data in transportation). *Transportation Research Board Conference*.

Caceres, H., Batta, R., & He, Q. (2017). School bus routing with stochastic demand and duration constraints. *Transportation Science*, *51*(4), 1349–1364. https://doi.org/10.1287/trsc.2016.0721

Cascetta, E., Nuzzolo, A., Russo, F., & Vitetta, A. (1996). A modified logit route choice model overcoming path overlapping problems. specification and some calibration results for interurban networks. *Transportation and Traffic Theory. Proceedings of The 13th International Symposium On Transportation And Traffic Theory, Lyon, France, 24-26 July 1996*.

Cats, O., & Gkioulou, Z. (2017). Modeling the impacts of public transport reliability and travel information on passengers' waiting-time uncertainty. *EURO Journal on Transportation and Logistics*, *6*(3), 247–270.

CBS. (2019). Wijk- en buurtkaart 2019 [Accessed: 2019-11-29].

Chen, A., Pravinvongvuth, S., Chootinan, P., Lee, M., & Recker, W. (2007). Strategies for selecting additional traffic counts for improving od trip table estimation. *Transportmetrica*, *3*(3), 191–211.

Chen, P., Shen, Q., & Childress, S. (2018). A gps data-based analysis of built environment influences on bicyclist route preferences. *International journal of sustainable transportation*, *12*(3), 218–231.

Chootinan, P., Chen, A., & Yang, H. (2005). A bi-objective traffic counting location problem for origin-destination trip table estimation. *Transportmetrica*, *1*(1), 65–80.

Chou, Y.-T. (2017). BATS and TBATS model. https://yintingchou.com/posts/bats-and-tbats-model/

Christiansen, P., Engebretsen, Ø., Fearnley, N., & Hanssen, J. U. (2017). Parking facilities and the built environment: Impacts on travel behaviour. *Transportation Research Part A: Policy and Practice*, *95*, 198–206.

Conway, M. W., Byrd, A., & van der Linden, M. (2017). Evidence-based transit and land use sketch planning using interactive accessibility methods on combined sched-

ule and headway-based networks. *Transportation Research Record*, *2653*(1), 45–53.

Cools, M., Moons, E., & Wets, G. (2009). Investigating the variability in daily traffic counts through use of arimax and sarimax models: Assessing the effect of holidays on two site locations. *Transportation Research Record*, *2136*(1), 57–66.

Cosslett, S. R. (1981). Efficient estimation of discrete-choice models. *Structural analysis of discrete data with econometric applications*, *3*, 51–111.

Cowpertwait, P. S. P., & Metcalfe, A. V. (2009). *Introductory time series with R*. Springer.

Curtis, C., & Scheurer, J. (2010). Planning for sustainable accessibility: Developing tools to aid discussion and decision-making. *Progress in Planning*, *74*(2), 53–106.

Dane, G., Feng, T., Luub, F., & Arentze, T. (2019). Route choice decisions of e-bike users: Analysis of gps tracking data in the netherlands. *International Conference on Geographic Information Science*, 109–124.

De Gruyter, C., Truong, L. T., & Taylor, E. J. (2020). Can high quality public transport support reduced car parking requirements for new residential apartments? *Journal of Transport Geography*, *82*, 102627.

de Freitas, L. M., Becker, H., Zimmermann, M., & Axhausen, K. W. (2019). Modelling intermodal travel in switzerland: A recursive logit approach. *Transportation Research Part A: Policy and Practice*, *119*, 200–213.

Delling, D., Pajor, T., & Werneck, R. F. (2014). Round-based public transit routing. *Transportation Science*, *49*(3), 591–604.

Delmelle, E. C., & Casas, I. (2012). Evaluating the spatial equity of bus rapid transit-based accessibility patterns in a developing country: The case of cali, colombia. *Transport Policy*, *20*, 36–46.

de Moraes Ramos, G., Mai, T., Daamen, W., Frejinger, E., & Hoogendoorn, S. (2020). Route choice behaviour and travel information in a congested network: Static and dynamic recursive models. *Transportation Research Part C: Emerging Technologies*, *114*, 681–693.

de Oña, J., & de Oña, R. (2015). Quality of service in public transport based on customer satisfaction surveys: A review and assessment of methodological approaches. *Transportation Science*, *49*(3), 605–622. https://doi.org/10.1287/trsc.2014.0544

Dibbelt, J., Pajor, T., Strasser, B., & Wagner, D. (2018). Connection scan algorithm. *Journal of Experimental Algorithmics*, *23*(1), 1–7.

Dijksma, S. (2020). *Amsterdam maakt ruimte*. Amsterdam Autoluw.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*(7), 2121–2159.

Ehlert, A., Bell, M. G., & Grosso, S. (2006). The optimisation of traffic count locations in road networks. *Transportation Research Part B: Methodological*, *40*(6), 460–479.

Esprabens, J., & Arango, A. (2020). Time series for beginners. https://bookdown.org/JakeEsprabens/431-Time-Series/modelling-time-series.html

ETH-Zurich. (2012). Position data processing.

Fan, J., Hu, Q., & Tang, Z. (2018). Predicting vacant parking space availability: An svr method with fruit fly optimisation. *IET Intelligent Transport Systems*, *12*(10), 1414–1420.

Farber, S., & Fu, L. (2017). Dynamic public transit accessibility using travel time cubes: Comparing the effects of infrastructure (dis) investments over time. *Computers, Environment and Urban Systems*, *62*, 30–40.

Fışkın, C. S., & Cerit, A. G. (2019). Forecasting domestic shipping demand of cement: Comparison of sarimax, ann and hybrid sarimax-ann. *2019 4th International Conference on Computer Science and Engineering (UBMK)*, 68–72.

Fosgerau, M., Frejinger, E., & Karlstrom, A. (2013). A link based network route choice model with unrestricted choice set. *Transportation Research Part B*, *56*, 70–80. https://doi.org/10.1016/j.trb.2013.07.012

Franklin, J. A. (2006). Recurrent neural networks for music computation. *INFORMS Journal on Computing*, *18*(3), 321–338. https://doi.org/10.1287/ijoc.1050.0131

Frejinger, E., Bierlaire, M., & Ben-Akiva, M. (2009). Sampling of alternatives for route choice modeling. *Transportation Research Part B: Methodological*, *43*(10), 984–994.

Friso, K., Wismans, L. J., & Tijink, M. B. (2017). Scalable data-driven short-term traffic prediction. *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 687–692.

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with lstm.

Ghanayim, M., & Bekhor, S. (2018). Modelling bicycle route choice using data from a gps-assisted household survey. *European Journal of Transport and Infrastructure Research*, *18*(2).

Golias, J., Yannis, G., & Harvatis, M. (2002). Off-street parking choice sensitivity. *Transportation Planning and Technology*, *25*(4), 333–348.

Grengs, J. (2004). Measuring change in small-scale transit accessibility with geographic information systems: Buffalo and rochester, new york. *Transportation Research Record*, *1887*(1), 10–17.

Grodi, R., Rawat, D. B., & Rios-Gutierrez, F. (2016). Smart parking: Parking occupancy monitoring and visualization system for smart cities. *SoutheastCon 2016*, 1–5.

Guevara, C. A., & Ben-Akiva, M. E. (2013). Sampling of alternatives in logit mixture models. *Transportation Research Part B: Methodological*, *58*, 185–198.

Guis, N., & Nijënstein, S. (2015). Modelleren van klantvoorkeuren in dienstregelingsstudies. *Colloquium Vervoersplanologisch Speurwerk. Antwerpen: NS*.

Hadas, Y., & Ranjitkar, P. (2012). Modeling public-transit connectivity with spatial quality-of-transfer measurements. *Journal of Transport Geography*, *22*, 137–147.

Halldórsdóttir, K., Rieser-Schüssler, N., Axhausen, K. W., Nielsen, O. A., & Prato, C. G. (2014). Efficiency of choice set generation techniques for bicycle routes. *European Journal of Transport and Infrastructure Research*, *14*(4), 332–348.

Hayashi, Y., Hsieh, M.-H., & Setiono, R. (2010). Understanding consumer heterogeneity: A business intelligence application of neural networks. *Knowledge-Based Systems*, *23*(8), 856–863.

Hillel, T., Bierlaire, M., & Jin, Y. (2019). *A systematic review of machine learning methodologies for modelling passenger mode choice* (tech. rep.). Technical Report TRANSP-OR 191025. EPFL.

Hirakawa, T., Yamashita, T., Tamaki, T., Fujiyoshi, H., Umezu, Y., Takeuchi, I., Matsumoto, S., & Yoda, K. (2018). Can ai predict animal movements? filling gaps in animal trajectories using inverse reinforcement learning. *Ecosphere*, *9*(10).

Hood, J., Sall, E., & Charlton, B. (2011). A gps-based bicycle route choice model for san francisco, california. *Transportation letters*, *3*(1), 63–75.

Hoogendoorn, M., & Funk, B. (2018). Machine learning for the quantified self. *On the art of learning from sensory data*.

Horni, A., Nagel, K., & Axhausen, K. W. (2016). *The multi-agent transport simulation matsim*. Ubiquity Press London.

Hyndman, R., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). *Forecasting with exponential smoothing: The state space approach*. Springer Science & Business Media.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice*. OTexts.

Jakob, M., Menendez, M., & Cao, J. (2020). A dynamic macroscopic parking pricing and decision model. *Transportmetrica B: Transport Dynamics*, *8*(1), 307–331.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical review*, *106*(4), 620.

Jin, J. G., Teo, K. M., & Odoni, A. R. (2016). Optimizing bus bridging services in response to disruptions of urban transit rail networks. *Transportation Science*, *50*(3), 790–804. https://doi.org/10.1287/trsc.2014.0577

JJ. (2016). MAE and RMSE - Which metric is better? https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), 20150202.

Kaplan, L. M., Hayes Jr, H. R., Devries, S., & Lindsay, M. G. (2009). Method of collecting parking availability information for a geographic database for use with a navigation system [US Patent 7,538,690].

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1725–1732.

Kazagli, E., Bierlaire, M., & Flötteröd, G. (2016). Revisiting the route choice problem: A modeling framework based on mental representations. *Journal of choice modelling*, *19*, 1–23.

Kelly, J. A., & Clinch, J. P. (2006). Influence of varied parking tariffs on parking occupancy levels by trip purpose. *Transport Policy*, *13*(6), 487–495.

Kennisinstituut voor Mobiliteit. (2019). *Mobiliteitsbeleid 2019*. Ministerie van Infrastructuur en Waterstaat. https://www.kimnet.nl/publicaties/rapporten/2019/11/12/mobiliteitsbeeld-2019-vooral-het-gebruik-van-de-trein-neemt-toe

Khaliq, A., van der Waerden, P., & Janssens, D. (2018). Modeling car drivers' on-street parking decisions using the integrated hierarchical information integration approach. *Transportation Research Record*, *2672*(49), 23–33.

Khandelwal, R. (2020). Time series prediction using SARIMAX. https://medium.com/datadriveninvestor/time-series-prediction-using-sarimax-a6604f258c56

Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Knapen, L., Hartman, I. B.-A., & Bellemans, T. (2017). Using path decomposition enumeration to enhance route choice models. *Future Generation Computer Systems*, –. https://doi.org/10.1016/j.future.2017.12.053

Knapen, L., Hartman, I. B.-A., Schulz, D., Bellemans, T., Janssens, D., & Wets, G. (2016). Determining structural route components from gps traces. *Transportation Research Part B: Methodological*, *90*, 156–171.

Knoppers, P., & Muller, T. (1995). Optimized transfer opportunities in public transport. *Transportation Science*, *29*(1), 101–105. https://doi.org/10.1287/trsc.29.1.101

Koch, T., Knapen, L., & Dugundji, E. (2019). Path complexity and bicyclist route choice set quality assessment. *Personal and Ubiquitous Computing*, *25*, 1–13.

Koch, T., Knapen, L., & Dugundji, E. (2020). Door-to-door transit accessibility using pareto optimal range queries. *Procedia Computer Science*, *170*, 107–114.

Kodransky, M., & Hermann, G. (2010). *Europe's parking u-turn: From accomodation to regulation*. Institute for Transportation; Development Policy.

Krizek, K., El-Geneidy, A., Iacono, M., & Horning, J. (2007). Access to destinations: Refining methods for calculating non-auto travel times.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://search-ebscohost-com.proxy.uba.uva.nl:2443/login.aspx?direct=true&db=buh&AN=123446102&site=ehost-live&scope=site

Kruyswijk, R. K. e. M. (2018). Parkeertarieven fors omhoog: Stijgingen tot 100 procent. https://www.parool.nl/nieuws/parkeertarieven-fors-omhoog-stijgingen-tot-100-procent~b42b4ee9

Kujala, R., Weckström, C., Mladenović, M. N., & Saramäki, J. (2018). Travel times and transfers in public transport: Comprehensive accessibility analysis based on pareto-optimal journeys. *Computers, Environment and Urban Systems*, *67*, 41–54.

Lawrence, S., Giles, C. L., Ah Chung Tsoi, & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, *8*(1), 98–113.

LeCun, Y., & Bengio, Y. (1998). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* (pp. 255–258). MIT Press.

Lei, C., & Ouyang, Y. (2017). Dynamic pricing and reservation for intelligent urban parking management. *Transportation Research Part C: Emerging Technologies*, *77*, 226–244.

Li, T., Chan, A. B., & Chun, A. H. (2010). Automatic musical pattern feature extraction using convolutional neural network. *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2010)*, 1x1.

Li, Z., & Xu, W. A. (2019). Path decision modelling for passengers in the urban rail transit hub under the guidance of traffic signs. *Journal of Ambient Intelligence and Humanized Computing*, *10*(1), 365–372.

Lin, L., Wang, Q., & Sadek, A. W. (2013). Short-term forecasting of traffic volume: Evaluating models based on multiple data sets and data diagnosis measures. *Transportation Research Record*, *2392*(1), 40–47.

Liu, Q., Wang, J., Chen, P., & Xiao, Z. (2017). How does parking interplay with the built environment and affect automobile commuting in high-density cities? a case study in china. *Urban Studies*, *54*(14), 3299–3317.

Long, T. (2019). Research on application of athlete gesture tracking algorithms based on deep learning. *Journal of Ambient Intelligence and Humanized Computing*, 1–9.

Ma, T., Antoniou, C., & Toledo, T. (2020). Hybrid machine learning algorithm and statistical time series model for network-wide traffic forecast. *Transportation Research Part C: Emerging Technologies*, *111*, 352–372.

Ma, X., Zhang, J., Du, B., Ding, C., & Sun, L. (2019). Parallel architecture of convolutional bi-directional lstm neural networks for network-wide metro ridership prediction. *IEEE Transactions on Intelligent Transportation Systems*, *20*(6), 2278–2288.

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, *17*(4).

Mai, T., Bastin, F., & Frejinger, E. (2018). A decomposition method for estimating recursive logit based route choice models. *EURO Journal on Transportation and Logistics*, *7*(3), 253–275.

Mai, T., Chan, K., & Jaillet, P. (2019). Generalized maximum causal entropy for inverse reinforcement learning. *arXiv preprint arXiv:1911.06928*.

Mai, T., Fosgerau, M., & Frejinger, E. (2015a). A nested recursive logit model for route choice analysis. *Transportation Research Part B: Methodological*, *75*(Supplement C), 100–112. https://doi.org/10.1016/j.trb.2015.03.015

Mai, T., Fosgerau, M., & Frejinger, E. (2015b). A nested recursive logit model for route choice analysis. *Transportation Research Part B: Methodological*, *75*, 100–112.

Mai, T., Nguyen, Q. P., Low, K. H., & Jaillet, P. (2019). Inverse reinforcement learning with missing data. *arXiv preprint arXiv:1911.06930*.

Manville, M., Shoup, D. C. et al. (2010). *Parking requirements as a barrier to housing development: Regulation and reform in los angeles*. Citeseer.

McFadden, D. (1973). Conditional logit analysis of qualitative choice behavior. *Frontiers of Econometrics, ed. P. Zarembka*.

Menghini, G., Carrasco, N., Schüssler, N., & Axhausen, K. W. (2010). Route choice of cyclists in zurich. *Transportation research part A: policy and practice*, *44*(9), 754–765.

Mo, M. (2019). *Cyclist's route choice prediction with inverse reinforcement learning* (Master's thesis). Universiteit van Amsterdam. the Netherlands.

Morikawa, T. (1989). *Incorporating stated preference data in travel demand analysis* (Doctoral dissertation). Massachusetts Institute of Technology.

Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(1), 86–97.

Nassir, N., Hickman, M., Malekzadeh, A., & Irannezhad, E. (2016). A utility-based travel impedance measure for public transit network accessibility. *Transportation Research Part A: Policy and Practice*, *88*, 26–39.

Nationaal Dataportaal Wegverkeer. (2021). Nationaal dataportaal wegverkeer. https://ndw.nu

Nguyen, Q. P., Low, B. K. H., & Jaillet, P. (2015). Inverse reinforcement learning with locally consistent reward functions. *Advances in Neural Information Processing Systems*, 1747–1755.

NHS England and NHS Improvement. (2020). Advanced forecast techniques. *NHS*. https://www.england.nhs.uk/wp-content/uploads/2020/01/advanced-forecasting-techniques.pdf

Nielsen, O. A. (2000). A stochastic transit assignment model considering differences in passengers utility functions. *Transportation Research Part B: Methodological*, *34*(5), 377–402.

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. *IEEE Conference on Computer Vision and Pattern Recognition*. https://hal.inria.fr/hal-00911179

O'Sullivan, D., Morrison, A., & Shearer, J. (2000). Using desktop gis for the investigation of accessibility by public transport: An isochrone approach. *International Journal of Geographical Information Science*, *14*(1), 85–104.

OVapi. (2020). Gtfs netherlands [Accessed: 2019-11-29].

Owen, A., & Levinson, D. M. (2015). Modeling the commute mode share of transit using continuous accessibility to jobs. *Transportation Research Part A: Policy and Practice*, *74*, 110–122.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pentaho. (2020). Time-series forecasting using TBATS model. https://blog.tenthplanet.in/time-series-forecasting-tbats/

Prato, C., & Bekhor, S. (2006). Applying branch-and-bound technique to route choice set generation. *Transportation Research Record: Journal of the Transportation Research Board*, (1985), 19–28.

Prato, C., & Bekhor, S. (2007). Modeling route choice behavior: How relevant is the composition of choice set? *Transportation Research Record: Journal of the Transportation Research Board*, (2003), 64–73.

Prato, C. G., Halldórsdóttir, K., & Nielsen, O. A. (2018). Evaluation of land-use and transport network effects on cyclists' route choices in the copenhagen region in value-of-distance space. *International Journal of Sustainable Transportation*, *12*(10), 770–781.

Rahman, F. I. (2020). Short term traffic flow prediction using machine learning-knn, svm and ann with weather information. *International Journal for Traffic & Transport Engineering*, *10*(3).

Rail Delivery Group. (2009). Passenger demand forecasting handbook.

Raveau, S., Guo, Z., Muñoz, J. C., & Wilson, N. H. (2014). A behavioural comparison of route choice on metro networks: Time, transfers, crowding, topology and socio-demographics. *Transportation Research Part A: Policy and Practice*, *66*, 185–195.

Reumers, S., Liu, F., Janssens, D., Cools, M., & Wets, G. (2013). Semantic annotation of global positioning system traces: Activity type inference. *Transportation Research Record*, *2383*(1), 35–43.

Rieser-Schüssler, N., Balmer, M., & Axhausen, K. W. (2013). Route choice sets for very high-resolution data. *Transportmetrica A: Transport Science*, *9*(9), 825–845.

Rosenblum, J., Hudson, A. W., & Ben-Joseph, E. (2020). Parking futures: An international review of trends and speculation. *Land Use Policy*, *91*, 104054.

Salonen, M., & Toivonen, T. (2013). Modelling travel time in urban networks: Comparable measures for private car and public transport. *Journal of Transport Geography*, *31*, 143–153.

Schakenbos, R., La Paix, L., Nijenstein, S., & Geurs, K. T. (2016). Valuation of a transfer in a multimodal public transport trip. *Transport Policy*, *46*, 72–81.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, *42*(3), 1–21.

Sener, I. N., Eluru, N., & Bhat, C. R. (2009). An analysis of bicycle route choice preferences in texas, us. *Transportation*, *36*(5), 511–539.

Shekhar, S., & Williams, B. M. (2007). Adaptive seasonal time series models for forecasting short-term traffic flow. *Transportation Research Record*, *2024*(1), 116–125.

Skorupa, G. (2019). Forecasting time series with multiple seasonalities using TBATS in Python. %7Bhttps://medium.com/intive-developers/forecasting-time-series-with-multiple-seasonalities-using-tbats-in-python-398a00ac0e8a%7D

Steinley, D. (2006). K-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, *59*(1), 1–34.

Stolfi, D. H., Alba, E., & Yao, X. (2017). Predicting car park occupancy rates in smart cities. *International Conference on Smart Cities*, 107–117.

Tenkanen, H., Heikinheimo, V., Järv, O., Salonen, M., & Toivonen, T. (2016). Open data for accessibility and travel time analyses: Helsinki region travel time and co2 matrix. *Geospatial data in a changing world: The short papers and poster papers of the 19th agile conference on geographic information science, 14–17 June 2016, Helsinki, Finland*.

Thakur, D., & Biswas, S. (2020). Smartphone based human activity monitoring and recognition using ml and dl: A comprehensive survey. *Journal of Ambient Intelligence and Humanized Computing*, 1–12.

Tian, Q., Yang, L., Wang, C., & Huang, H.-J. (2018). Dynamic pricing for reservation-based parking system: A revenue management method. *Transport Policy*, *71*, 36–44.

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, *4*(2), 26–31.

Ton, D., Cats, O., Duives, D., & Hoogendoorn, S. (2017). How do people cycle in amsterdam, netherlands?: Estimating cyclists' route choice determinants with gps data from an urban area. *Transportation Research Record*, *2662*(1), 75–82.

Ton, D., Duives, D., Cats, O., & Hoogendoorn, S. (2018). Evaluating a data-driven approach for choice set identification using gps bicycle route choice data from amsterdam. *Travel Behaviour and Society*, *13*, 105–117.

Tribby, C. P., & Zandbergen, P. A. (2012). High-resolution spatio-temporal modeling of public transit accessibility. *Applied Geography*, *34*, 345–355.

Tutorialspoint. (2021). Time series - Walk forward validation. https://www.tutorialspoint.com/time%5C_series/time%5C_series%5C_walk%5C_forward%5C_%20validation.htm

US Department of Transportation. (2018). How do weather events impact roads? https://ops.fhwa.dot.gov/weather/q1%5C_roadimpact.htm

van der Waerden, P., & Oppewal, H. (1996). Modelling the combined choice of parking lot and shopping destimation. *World transport research: proceedings of 7th world conference on Transport Research: volume 1: travel behaviour*, 129–137.

Van Wee, B., & Handy, S. (2016). Key research themes on urban space, scale, and sustainable urban mobility. *International journal of sustainable transportation*, *10*(1), 18–24.

van Cranenburgh, S., & Alwosheel, A. (2019). An artificial neural network based approach to investigate travellers' decision rules. *Transportation Research Part C: Emerging Technologies*, *98*, 152–166.

Vazifeh, M. M., Santi, P., Resta, G., Strogatz, S. H., & Ratti, C. (2018). Addressing the minimum fleet problem in on-demand urban mobility. *Nature*, *557*(7706), 534–538.

Vlahogianni, E. I., & Karlaftis, M. G. (2013). Testing and comparing neural network and statistical approaches for predicting transportation time series. *Transportation Research Record: Journal of the Transportation Research Board*, *2399*(1), 9–22. https://doi.org/10.3141/2399-02

Voerknecht, H. (2016). De wegen slibben dicht. hoe komt CROW tot die conclusie? https://www.crow.nl/kennis/bibliotheek-verkeer-en-vervoer/kennisdocumenten/de-wegen-slibben-dicht,-hoe-komt-crow-aan-die-info

Vythoulkas, P. C., & Koutsopoulos, H. N. (2003). Modeling discrete choice behavior using concepts from fuzzy set theory, approximate reasoning and neural networks. *Transportation Research Part C: Emerging Technologies*, *11*(1), 51–73.

Wang, S., Mo, B., & Zhao, J. (2020). Deep neural networks for choice analysis: Architecture design with alternative-specific utility functions. *Transportation Research Part C: Emerging Technologies*, *112*, 234–251.

Wang, S., Wang, Q., & Zhao, J. (2020). Multitask learning deep neural networks to combine revealed and stated preference data. *Journal of Choice Modelling*, *37*, 100236.

Wang, S., & Zhao, J. (2019). *An empirical study of using deep neural network to analyze travel mode choice with interpretable economic information* (tech. rep.). Massachusetts Institute of Technology.

Wardenier, N., Knapen, L., Koch, T., & Dugundji, E. (2019). Improving bicycle route choice set generation using route complexity in GPS traces. *Proceedings of the 98th Annual Meeting of the Transportation Research Board*.

Weisstein, E. W. (2020). Fourier series. https://mathworld.wolfram.com/FourierSeries.html

Welch, G., Bishop, G. et al. (1995). An introduction to the kalman filter.

What is screenline analysis? [Accessed: 2017-07-05]. (n.d.). %5Curl%7Bhttp://www.caliper.com/glossary/what-is-screenline-analysis.htm%7D

Wu, G., Ding, Y., Li, Y., Luo, J., Zhang, F., & Fu, J. (2017). Data-driven inverse learning of passenger preferences in urban public transits. *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 5068–5073.

Wu, G., Li, Y., Bao, J., Zheng, Y., Ye, J., & Luo, J. (2018). Human-centric urban transit evaluation and planning. *2018 IEEE International Conference on Data Mining (ICDM)*, 547–556.

Xie, Y., Zhao, K., Sun, Y., & Chen, D. (2010). Gaussian processes for short-term traffic volume forecasting. *Transportation Research Record*, *2165*(1), 69–78.

Yang, C., & Liu, Q. (2009). Enhanced genetic algorithm for selecting traffic counting location. *Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on*, 1–6.

Yang, H., Yang, C., & Gan, L. (2006). Models and algorithms for the screen line-based traffic-counting location problems. *Computers & Operations Research*, *33*(3), 836–858.

Yang, H., & Zhou, J. (1998). Optimal traffic counting locations for origin–destination matrix estimation. *Transportation Research Part B: Methodological*, *32*(2), 109–126.

Yang, S., Ma, W., Pi, X., & Qian, S. (2019). A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. *Transportation Research Part C: Emerging Technologies*, *107*, 248–265.

Yang, Y., Zheng, Z., & E, W. (2020). Interpretable neural networks for panel data analysis in economics.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 3320–3328.

Zhang, J., Ma, X., Ding, C., Wang, Y., & Liu, J. (2018). *Forecasting subway demand in large-scale networks: A deep learning approach* (tech. rep.).

Zhou, T., Han, G., Xu, X., Lin, Z., Han, C., Huang, Y., & Qin, J. (2017). δ-agree adaboost stacked autoencoder for short-term traffic flow forecasting. *Neurocomputing*, *247*, 31–38.

Zhu, X., Guo, J., Huang, W., Yu, F., & Brian Park, B. (2018). Real time short-term forecasting method of remaining parking space in urban parking guidance systems. *Promet-Traffic&Transportation*, *30*(2), 173–185.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. *AAAI*, *8*, 1433–1438.

Zimmermann, M., & Frejinger, E. (2020). A tutorial on recursive models for analyzing and predicting path choice behavior. *EURO Journal on Transportation and Logistics*, 100004.

Zimmermann, M., Mai, T., & Frejinger, E. (2017). Bike route choice modeling using gps data without choice sets of paths. *Transportation research part C: emerging technologies*, *75*, 183–196.

ACKNOWLEDGEMENTS

This research has been conducted in the framework of the North/South-line Impact Study, funded in part by the Municipality of Amsterdam and the regional transportation authority of Amsterdam as part of a public private partnership with CWI. I would like to thank my supervisors Elenna Dugundji and Rob van der Mei for their support in all matters throughout the research.

I would gratefully like to acknowledge Machiel Kouwenberg and others at the Vervoerregio for support in facilitating practicalities of the panel data collection via mobile app with Marcel Bijlsma at Mobidot and Matthijs de Gier at Kantar Public, as well as Rutger Veldhuizen van Zanten and Barry Ubbels at the Municipality for their support in facilitating practicalities of the public transit smart card data collection with TLS. Additionally, Rogier van der Groep at the Municipality served a key role as liaison with the Department of Research, Information and Statistics. I would also like to thank my fellow academic work group researchers: Ties Brands, Malvika Dixit, Ori Ruben, Paul Koster, Eric Koomen, Jip Claessens, Jan Rouwendal.

I would gratefully like to acknowledge the Vrije Universiteit Amsterdam and CWI for support in the design and implementation of the computing infrastructure. The following persons deserve specific mention: Rudi Bergval, Peter Stol, Kees Bot at the IT for Research team at VU; Erik Koedam, Will Tummers and Tim van Loon at the VU Central IT department; Tom Paffen, Petra Tolen and Martijn de Graaf at the VU legal affairs office; and Jolien Scholten and Mark Bruyneel at the VU university library research data management team, as well as Arjen de Rijke, Niels Nes and Carl Schulz at the IT and Facilities department of CWI.

With respect to chapter 3, I would like to thank Prof. Ger Koole, head of the Analytics and Optimization group at the Department of Mathematics at the Vrije Universiteit Amsterdam for giving me the opportunity to initially work on the research during the Project Operations of Business Processes (POBP) together with Elisabeth Fokker, who subsequently joined the North/South-line Impact Study team at CWI. Hereby I also thank VU POBP team members Sammy Eshun, Alex Korthouwer and Shwetambari Tiwari for their contribution to the study design, analysis of results, and interpretation of results. I would like to thank Bas Schotten, Abdel el Nali, Marco van Leeuwen and Josephine Chew at the Parking Department of the Municipality.

With respect to chapters 4 and 5, I would like to thank Prof. Bert Zwart, head of the Stochastics Group at CWI for giving me the opportunity to collaborate with Sergey Khalil and Ruurd Buijs during internships that they respectively carried out at CWI.

With respect to chapter 6, I would like to acknowledge the Traffic and Public Space Department of the Municipality of Amsterdam for their support, with special thanks to Bas Bussink, Eric de Kievit, Laurens Peijs and Stefan Verduin.

With respect to chapter 7, I am again grateful to Prof. Ger Koole and the Department of Mathematics and the Faculty of Science at the Vrije Universiteit Amsterdam for giving

me the opportunity to work on the research during the Project Operations of Business Processes (POBP) together with Tom de Valk, Stan van Loon, Bas van der Bijl, Bart Gijsbertsen and Yorran Reurich. I am grateful to the Ministry of Infrastructure and Water Management and the Nationaal Dataportaal Wegverkeer for facilitating access to motorway traffic counter data.

With respect to chapter 8, I would especially like to thank Luk Knapen for useful discussions related to the research topic and assistance with the data preparation. The research in this chapter has been made possible in part by guest researcher funding for Luk Knapen received from "Stochastics - Theoretical and Applied Research" (STAR) in the Netherlands.

In closing, I would like to thank my parents and my sister for their support.

Thomas Koch
Lelystad, March 2022

SAMENVATTING

Dit proefschrift is het resultaat van onderzoek in het kader van de impactstudie Noord/Zuidlijn, waar Centrum Wiskunde & Informatica, onderzoeksgroep Stochastiek, trekker was van het thema rond Mobiliteit en Bereikbaarheid. De opening van de Noord/Zuidlijn in juli 2018 was een grote verandering voor het openbaar vervoer netwerk van Amsterdam. Ons onderzoek had de focus op de veranderingen in multimodale ketenverplaatsingen met betrekking tot openbaar vervoer en lopen, en de effecten op parkeergarages en P+R locaties. Daarnaast hebben wij ook gekeken naar de effecten op intensiteiten van autoverkeer en de routekeuze van fietsers.

OPENBAAR VERVOER EN LOPEN

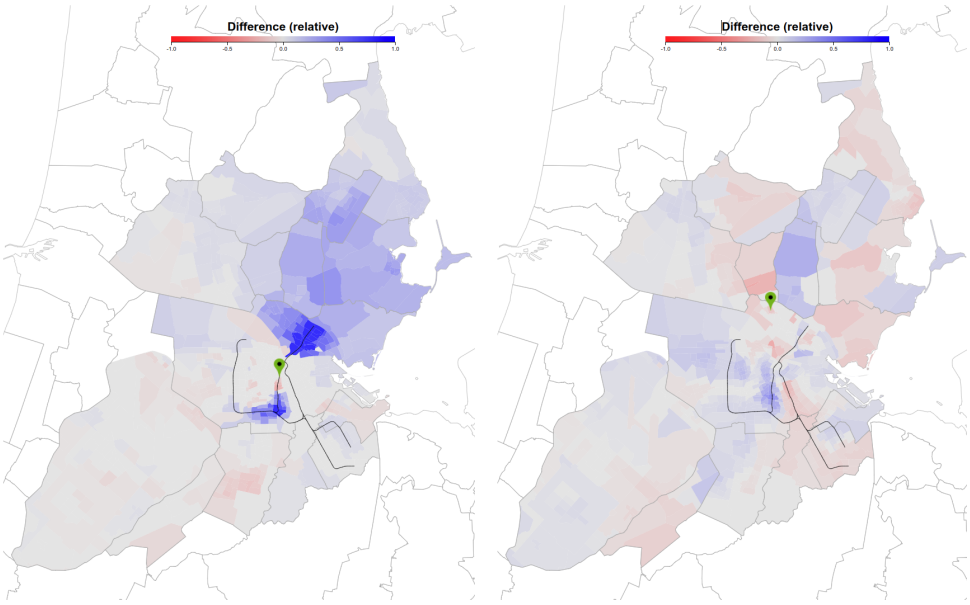
Een van de resultaten uit ons onderzoek is dat ,zoals verwacht, vanuit het centrum de reistijd richting het noorden en zuiden flink is afgenomen. Omdat niet iedereen dicht bij een bushalte of een metrostation kan wonen, in ons onderzoek kijken we verder dan de reistijd van halte tot halte. Daarom selecteren we steekproefsgewijs adressen totdat de hele buurt binnen 300 meter van een geselecteerde herkomst of bestemming ligt. De reistijd die wij berekenen is het gewogen gemiddelde van de reistijden tussen elk punt in beide buurten, met als gewicht hoeveel adressen dichtbij dat punt liggen.

Echter voor een reiziger in het openbaar vervoer is niet alleen de reistijd een factor maar ook de frequentie van deze verbinding. Een reiziger heeft meer aan een langzame verbinding die elke tien minuten rijdt dan een snelle verbinding die om het uur rijdt. Om deze effecten te kunnen bepalen gebruiken we de 'rooftopmethode'. Hierbij kijken we voor elke minuut in een tijdsperiode, zoals de ochtendspits, wat de eerste aankomsttijd is. Op die manier kijken we niet alleen naar de reistijd deur tot deur maar ook hoe lang je moet wachten tussen elke verbinding. Het gemiddelde reistijd plus wachttijd over de gehele tijdsperiode, is een goede indicatie die een reiziger kwijt is, om op tijd aan te komen voor een afspraak op zijn of haar bestemming.

Een interessante uitkomst van dit onderzoek was dat er op de verbinding tussen Amsterdam Molenwijk en het Amstelkwartier in reistijd op zich, geen groot verschil te zien is, maar dat het effect van de lagere frequenties door de IJtunnel en eventuele extra overstap wel zichtbaar was in de berekening met de rooftop-methode. De resultaten van dit onderzoek zijn ook te bekijken met een interactieve visualisatie; twee voorbeelden zijn te zien in Figuur 9.9

PARKEREN EN P+R

Verder hebben we autoverkeer en parkeren onderzocht. In dit proefschrift kijken we met tijdreeksmodellen naar een dataset met het aantal vrije parkeerplekken in parkeergarages en P+R locaties tussen 2018 en 2020. Uit deze modellen blijkt dat met factoren als evenementen, neerslag, onweer, windsnelheden, temperatuur, zonneschijn en de opening van de Noord-Zuidlijn, het aantal vrije plekken beter te voorspellen is. De opening



(a) Vanaf Reguliersbuurt in Centrum

(b) Vanaf Molenwijk in Noord

Figuur 9.9: Verschil in reistijden deur tot deur met openbaar vervoer. Blauw is een verbetering van de reistijd na de opening van de Noord/Zuidlijn en rood een verslechtering.

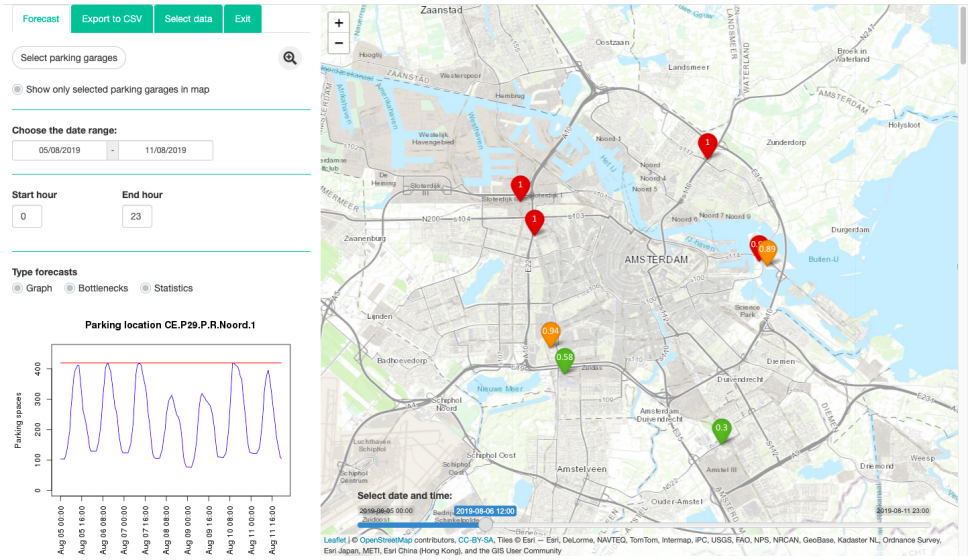
van de Noord/Zuidlijn had dus een significant maar klein effect op het aantal bezette parkeerplekken in garages en P+R's in Amsterdam. De resultaten uit dit onderzoek zijn gepresenteerd op basis van een decision support system, weergegeven in Figuur 9.10. Op de kaart is de fractie van de parkeerbezetting zichtbaar voor de acht P+R locaties op dinsdag 6 augustus 2019 rond 12:00.

AUTO

Een soortgelijke analyse hebben we uitgevoerd op data met verkeerstellingen uit de Nationale Databank Wegverkeer, om het effect van de opening van de Noord/Zuidlijn en de COVID-19 pandemie te bestuderen. Het wegverkeer in de IJtunnel is verminderd na de opening van de Noord/Zuidlijn. Wederom zijn de voorspellingen en visualisaties te bekijken in een interactief dashboard (Figuur 9.11)

FIETS

Tenslotte kan je voor een onderzoek naar mobiliteit in Amsterdam niet om de fiets heen: een derde van de verplaatsingen door Amsterdammers is verricht op de fiets. In dit promotieonderzoek is gekeken naar verschillende methoden, om een beter inzicht te krijgen in het routekeuze gedrag van fietsers. Dit met als doel om betere verkeersmodellen en -simulaties rond de impact van de Noord/Zuidlijn, en ook toekomstige projecten zoals Sprong over het IJ, te kunnen bouwen.



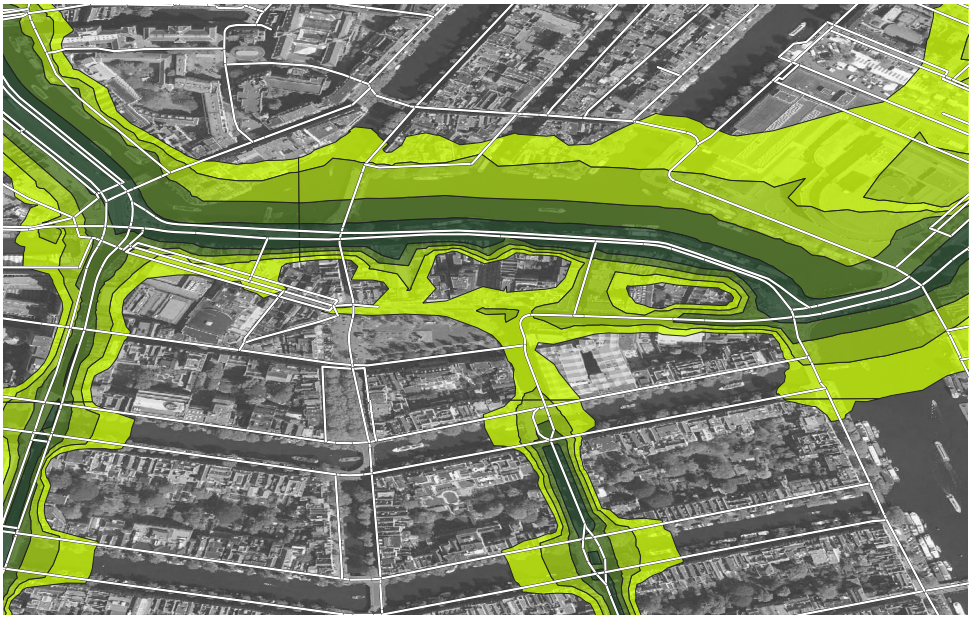
Figuur 9.10: Dashboard parkeren. Linksonder is de parkeerbezetting van P+R Noord gevisualiseerd in de week van maandag 5 augustus tot en met zondag 11 augustus 2019.

Figuur 9.11: Dashboard wegverkeer. De grafiek beschrijft de verkeersintensiteit in de IJtunnel in de periode januari 2018 tot en met december 2019. Andere verkeerslocaties zijn te kiezen met de tab 'Map'.

Het begin van dit onderzoek was een studie naar het toepassen van Route Complexity op de Fietstelweek dataset. Met route complexity knip je de gekozen route in kleine stukjes totdat elk stukje een kortste pad is. Het idee daarachter is dat men via verschillende herkenningspunten fietst en dat men realistischere fietsroutes kan generen als je route complexity meeneemt als attribuut van de generatie. De conclusie van dit onderzoek was dat fiets meer complexere route-keuzes maken dan automobilisten.

Hierna hebben we geprobeerd om een routekeuze model te schatten op basis van de fietstelweek data. Onze eerste poging met inverse reinforcement learning en recursive logit strandde nadat bleek dat de complexiteit van het fietsnetwerk en het aantal keuzemogelijkheden, te hoog was om dit toe te kunnen passen in Amsterdam.

Uiteindelijk hadden we het meeste succes met een klassiek keuzemodel, de zogenaamde mixed logit met de path size logit. Hieruit bleek dat er een grote keuze variëteit is in hoeveel mensen bereid zijn om langer te fietsen, maar ook dat er zoals verwacht er een voorkeur bestaat voor groene gebieden, langs het water en/of over fietspaden te fietsen. De nabijheid van een weg met luid wegverkeer, blijkt een negatieve factor in routekeuze.



Figuur 9.12: Kaart van het geluidsemissie-model dat gebruikt is om het effect op routekeuze van drukke wegen te bepalen.

CONCLUSIE

Uiteindelijk kunnen de methodologische ontwikkelingen van dit proefschrift in combinatie met een synthetische populatie en activiteiten generatie ingezet worden om de verplaatsingen per vervoersmodaliteit te modelleren. Uit dit model kunnen vervolgens zaken als congestie worden afgeleid.

PUBLICATIONS OF THE AUTHOR

Peer reviewed publications

1. **Thomas Koch**, Rob van der Mei and Elenna Dugundji. "The optimization of traffic count locations in multi-modal networks."Procedia computer science 130 (2018): 287-293.

2. **Thomas Koch**, Luk Knapen and Elenna Dugundji. "Path complexity for observed and predicted bicyclist routes."Procedia Computer Science 151 (2019): 393-400.

3. Luk Knapen, **Thomas Koch** and Elenna Dugundji. "Bicyclist route choice: Data exploration and research project outline."Procedia Computer Science 151 (2019): 401-408.

4. Maurits Waterbolk, Jasper Tump, Rianne Klaver, Rosalie van der Woude, Daniel Velleman, Joost Zuidema, **Thomas Koch** and Elenna Dugundji. "Detection of ships at mooring dolphins with hidden Markov models."Transportation Research Record 2673.4 (2019): 439-447.

5. **Thomas Koch**, Luk Knapen and Elenna Dugundji. "Door-to-door transit accessibility using Pareto optimal range queries."Procedia Computer Science 170 (2020): 107-114.

6. Ruurd Buijs, **Thomas Koch** and Elenna Dugundji. "Using neural nets to predict transportation mode choice: an Amsterdam case study."Procedia Computer Science 170 (2020): 115-122.

7. **Thomas Koch** and Elenna Dugundji. "A review of methods to model route choice behavior of bicyclists: inverse reinforcement learning in spatial context and recursive logit."Proceedings of the 3rd ACM SIGSPATIAL International Workshop on GeoSpatial Simulation (2020): 30-37.

8. Ruurd Buijs, **Thomas Koch** and Elenna Dugundji. "Using neural nets to predict transportation mode choice: Amsterdam network change analysis."Journal of Ambient Intelligence and Humanized Computing 12.1 (2021): 121-135.

9. Elisabeth Fokker, **Thomas Koch** and Elenna Dugundji. "The impact of a new public transport line on parking behavior."Procedia Computer Science 184 (2021): 210-217.

10. Sergey Khalil, Chintan Amrit, **Thomas Koch** and Elenna Dugundji. "Forecasting public transport ridership: Management of information systems using CNN and LSTM architectures."Procedia Computer Science 184 (2021): 283-290.

11. Jan-Willem Feilzer, Daan Stroosnier, Elenna Dugundji and **Thomas Koch**. "Predicting lessee switch behavior using logit models."Procedia Computer Science 184 (2021): 380-387.

12. **Thomas Koch** and Elenna Dugundji. "Limitations of recursive logit for inverse reinforcement learning of bicycle route choice behavior in Amsterdam."Procedia Computer Science 184 (2021): 492-499.

13. Ruurd Buijs, **Thomas Koch** and Elenna Dugundji. "Applying transfer learning and various ANN architectures to predict transportation mode choice in Amsterdam."Procedia Computer Science 184 (2021): 532-540.

14. Elisabeth Fokker, **Thomas Koch** and Elenna Dugundji. "Long-term forecasting of off-street parking occupancy for smart cities". Proceedings of the 100th Annual Meeting of the Transportation Research Board (2021).

15. Britt Verhoeven, Nimo van Hout, Archika Devaraj, Hans Zwitzer, Terry Crapts, Andrei Ion, **Thomas Koch** and Elenna Dugundji. "Short-term forecasting of air cargo demand from a European airport hub to the United States during COVID-19."Proceedings of the 100th Annual Meeting of the Transportation Research Board (2021).

16. **Thomas Koch**, Luk Knapen and Elenna Dugundji. "Path complexity and bicyclist route choice set quality assessment."Personal and Ubiquitous Computing 25.1 (2021): 63-75.

17. Heleen Eisen, Joachim van der Lei, Joost Zuidema, **Thomas Koch** and Elenna Dugundji. "An evaluation of first-come, first-served scheduling in a geometrically-constrained wet bulk terminal."Frontiers in Future Transportation 2 (2021): 23.

18. Elisabeth Fokker, **Thomas Koch**, Marco van Leeuwen and Elenna Dugundji. "Short-term forecasting of off-street parking occupancy". Transportation Research Record 2676.1 (2021): 637-654.

19. **Thomas Koch**, and Elenna Dugundji. "Taste variation in environmental features of bicycle routes."Proceedings of the 14th ACM SIGSPATIAL International Workshop on Computational Transportation Science (2021): 1-10.

20. Morris Heijke, Bruno van Beek, Laurens Papadopoulos, Noa Heertje, **Thomas Koch** and Elenna Dugundji. "Predicting freight traffic flow using deep learning before and during road network maintenance"Proceedings of the 101th Annual Meeting of the Transportation Research Board (2022).

21. Bas van der Bijl, Bart Gijsbertsen, Stan Van Loon, Yorran Reurich, Tom de Valk, **Thomas Koch** and Elenna R. Dugundji. A comparison of approaches for the time series forecasting of motorway traffic flow Rate at hourly and daily aggregation levels. Accepted for ANT 2022: The 13th International Conference on Ambient Systems, Networks and Technologies (2022).

22. Annabel van den Berg, Lotte Bootsma, Thomas Bovenberg, Rosa Moerbeek, Eelco de Jong, Sergey Khalil, **Thomas Koch** and Elenna R. Dugundji. Year-ahead ambient temperature forecasting in pharmaceutical transport lanes thermal conditions. Accepted for ANT 2022: The 13th International Conference on Ambient Systems, Networks and Technologies (2022).