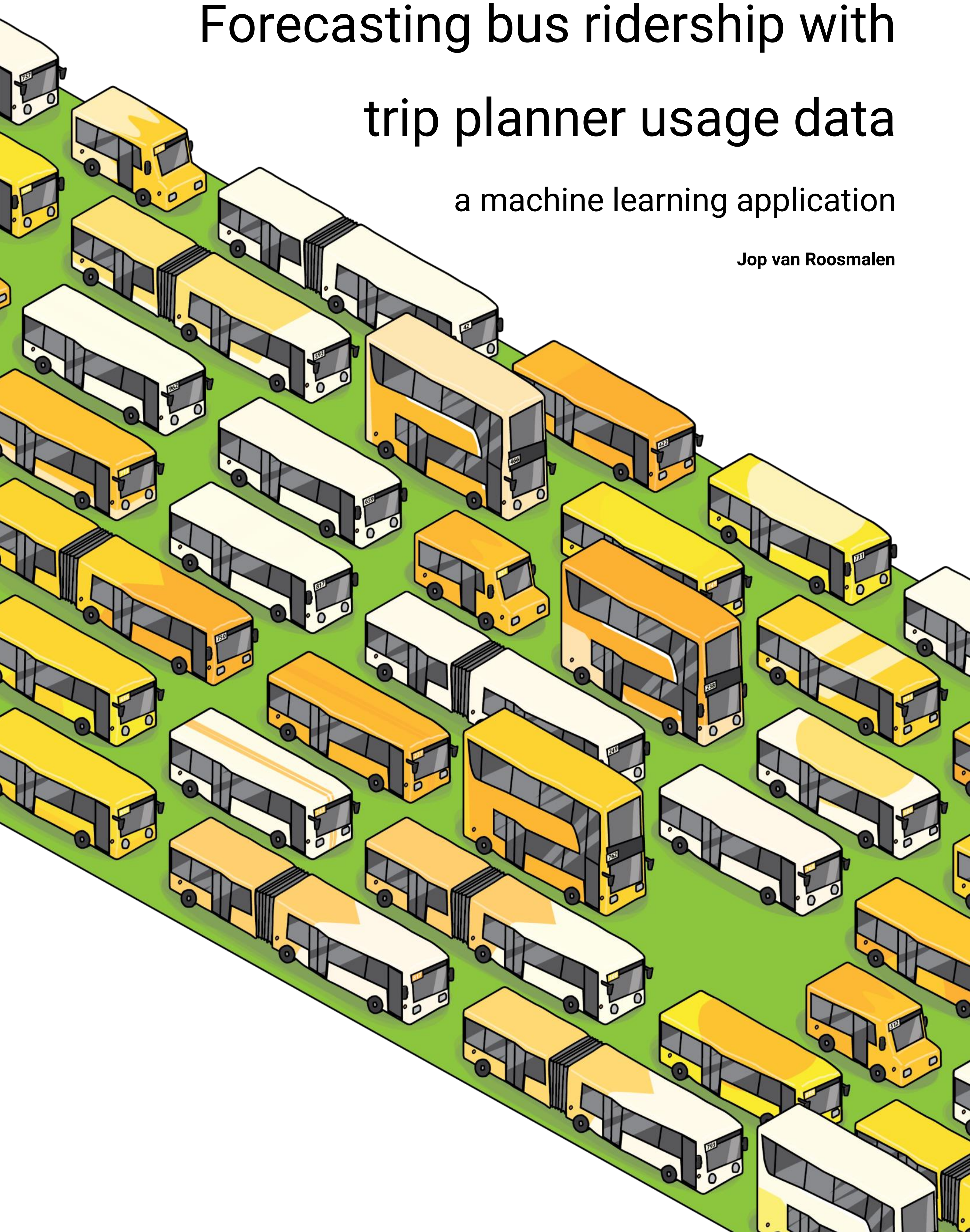
Forecasting bus ridership with trip planner usage data

a machine learning application

Jop van Roosmalen



MSc thesis, March 2019

Program Industrial Engineering & Management

Track Production & Logistics Management

University of Twente

Author

Jop van Roosmalen

Supervisors:

dr. Chintan Amrit - *University of Twente*

dr. Engin Topan - *University of Twente*

dr. Niels van Oort - *Delft University of Technology*

Stephan Metz - *OV-bureau Groningen Drenthe*

Roy van den Berg - *Translink*

Susan Zethof - *9292*

Management Summary

Public transport gives much attention to environmental impact, costs and traveler satisfaction. Good short-term demand forecasting models can help improve these performance indicators. It can help prevent denied boarding and overcrowding in buses by detecting insufficient capacity beforehand. It could be used to operate more economically by decreasing the frequency or the bus size if there is overcapacity. It could help operators plan their buses during incidental occasions like big public events where little information is known and it can finally be used to reliably inform the travelers on the current crowdedness (Ohler et al., 2017; Van Oort et al., 2015a; Pereira et al., 2015).

This study investigates the usefulness of a new data source; the usage data of a trip planner for public transport. In the Netherlands there are multiple planners available to help find the most optimal (multimodal) travel advice. These trip planners require a date, a time, an origin and a destination, based on which they are able to construct multiple alternative journeys from which the user can choose. The usage data of these planners could potentially be very insightful, the main research question therefore is: *Can one forecast short-term ridership of buses using data containing the consulted travel advices from a widely used trip planner for public transport and what accuracy can one achieve in different scenarios?*

Literature

During the literature review no research was found using trip planner usage data for forecasting public transport demand. However, we found multiple factors which are interesting to include. We will include factors from the groups: Temporal, Demand characteristics, Weather, Event, Holidays and Transit characteristics.

Case study

For the study we used data of 20 lines (urban and regional) operated by Qbuzz in Groningen and Drenthe for the first three months of 2017. The time period is too short to investigate holidays and large public events.

Data

For this study the data of 9292 was used. 9292 is one of the major trip planners in the Netherlands and includes all public transport modes for the whole country. A

regression analysis is used to determine the forecasting potential of the trip planner usage data. This data is regressed towards smart card transaction data.

A few challenges had to be overcome in order to perform the study. Firstly, the data that is logged by 9292 is not optimized to be used for forecasting demand: It is unknown if two requests are made by the same person (viewing an alternative journey plan is logged as a separate request) and there is no identifier for the bus trip stored (only a line number). It is also difficult to match the trip planner trips with bus trips, since, over time, the 9292 private bus stops database evolved differently and there is no information stored on the actual delay although they are used during the construction of the travel advices.

Secondly, everyone has his own strategy (for different scenarios) in planning a trip and will use the planner differently to fulfill his needs. The user interface design and functionality of the trip planner influence this behavior and therefore directly impact the usage data. Furthermore, it is unknown if a travel plan is made for one person or for a group of people.

Methodology

We developed a model for forecasting the number of people boarding and a model for forecasting the number of people alighting at a certain stop. These forecasts are defined at the vehicle-stop level. By counting the number of people boarding and subtracting the number of people alighting along the trip, the forecasted number of passengers after a stop can be calculated (Ohler et al., 2017).

We compare five different machine learning models: multiple linear regression, decision tree, random forests, neural networks and support vector regression with a radial basis kernel (Zhang et al., 2017; James et al., 2013). We compare these models with two simple rules: 1 predict the same number as last week, and 2 predict the historic average as number. The models are implemented in the Scikit-Learn library of Python (Pedregosa et al., 2011) and the data is stored in a PostgresSQL database.

The trip planner datasets and smart card dataset are merged and preprocessed. The resulted dataset is rather sparse; a lot of stops have zero passengers boarding or alighting or are not requested in the trip planner. Therefore we investigated if subsampling is needed. From the datasets useful data is selected and features are constructed. The features are standardized. Different number of features are tested, these features are selected based on recursive elimination using a simple random forests model. Finally, the hyperparameters of the models are tuned and the optimal configurations are stored. The scores are validated by using cross validation.

Results

We used the trips of one route during the morning peak to test our models. We used different kind of data partitions to train these models. All models are constructed with a planning horizon of 15 minutes. In most cases the best performing model used 20 features, the maximum number that was allowed.

The random forests model predicted the number of people boarding most accurate with a Root Mean Squared Error (RMSE) of 2.55 (R2 of 0.76). The random forests model forecasted the number of people alighting most accurate as well, with an RMSE of 2.20 (R2 of 0.76). The lower RMSE indicates that the number of people alighting is more predictable. In both cases the best version of the other models outperformed the forecasts of rule 1 and 2. It was discovered that subsampling had a slight negative effect.

When combining the boarding and alighting model, random forests outperforms the other machine learning models with an RMSE of 8.72. However, rule 2 has an RMSE of 8.603. When looking at the percentage of trips correctly forecasted within an absolute error of 5 passengers, rule 2 outperforms the random forests model with 84.08% against 58.9%. Thus, rule 2 outperforms the machine learning models when it comes to forecasting the number of passengers. Combining the best performing boarding and alighting model does not lead to the best forecast for the number of passengers. When looking at the percentage of correct maximum number of passengers predictions of trips – the most important indicator for adjusting the size of the bus –, the forecasts of rule 2 and the random forests model severely underestimated (more than 10 passengers lower as the real value) the maximum number of passengers for more than 27% of the trips.

The two most important features are the historical average of the number of people boarding (or alighting) and the number of requests for the same line aggregated over a window of 3 hours. The first feature was included to give an adequate baseline. The disaggregated version of the second feature is probably too noisy and fluctuates too much. Aggregating this feature over time helps to reveal the underlying trend more reliably.

Conclusion and recommendation

The trip planner usage data is an interesting source to detect the number of additional people boarding or alighting. Especially since this process could be fully automated. However, the different organizations should adjust their data structures in order to construct more useful features, do more valuable analysis and to streamline the whole data preprocessing process of merging the different datasets.

Researchers could help this process by further developing these forecasting models, testing more features and models, testing the models in different scenario's and by researching models that forecast the maximum number of passengers using a different method since combining the boarding and alighting model leads to interference errors.

Preface

This master thesis is the final stage of my master studies Industrial Engineering & Management with specialization Production and Logistics Management. It marks the end of my studies at two different universities.

I would like to thank Niels, Roy, Susan and Stephan who initiated the project and who gave me the opportunity to execute it. Furthermore, I would like to thank them, as well as Chintan and Engin, for their involvement, enthusiasm and feedback without which this project would not have been possible. I also want to thank all the people at OV-bureau Groningen Drenthe for a welcoming working environment.

I hope that this project contributes to the collaboration between OV-bureau Groningen Drenthe, Translink and 9292. And will contribute to better insights and new common practices.

Table of contents

Table of figures

1 Introduction

1 Introduction

1.1 Research motivation

One of the main challenges in public transport is matching transport demand and supply given budget constraints. A mismatch in demand and supply leads to extended travel times, delays and less comfort in the short-term, which can have effect on the mode choice in the long-term. Ohler, Krempels and Möbus (2017) and Oort et al. (2015a) present multiple reasons why good demand forecasting is important. For instance, forecasts can be used to allocate buses to prevent cramming in buses which results in more favorable travel conditions for travelers. By allocating buses where needed, no capacity will be wasted. Furthermore, this will prevent delays which are caused by extended alighting and boarding times during peak demands. Moreover, in the current high-tech era, travelers expect advanced accurate traffic information about expected arrival and departure times.

The fact that demand is fluctuating short term and long term, makes planning for sufficient capacity a complex task. For example, since travelers have different habits and activities over space and time, their need for public transport varies and is temporal and spatial dependent. Transit operators try to cope with fluctuations in demand by updating their network and timetable design one or two times a year. These reparations to the bus schedule involve high costs because of a snowball effect upon changes to underlying operations. During the year, some reinforcement buses are available at all time to be assigned if needed. However, insufficient supply is often detected too late. Sometimes this results in measures taken by traffic control, for example by sending (additional) buses. If insufficient supply could be forecasted, efficient matches in demand and supply could be realized. This can also avoid inconvenience for travelers.

To predict ridership and changes in demand, most operators do not have a multimodal transport model that matches the level of detail of the public transport operations (Van Oort et al., 2015b). Most operators use spreadsheets with simple rules instead of advanced traffic models. Over the last decade, a lot of research is conducted to improve public transport using tracking data like automatic vehicle location (AVL) data of buses, smart card data and mobile phone data on telephone mast level (see e.g. Van Oort et al., 2015c). Some of this research has been done to develop new ways of revealing transport demand. The trend in research towards big data is partly caused by the fact that these data are becoming more available and partly because public transport organizations want to operate most cost effectively while delivering a higher quality of services. Big data can help to solve these contradicting requirements (Van Oort et al., 2015a; Van Oort et al., 2015b).

A big data source that could be interesting for forecasting public transport demand is the usage data of trip planners for public transport. Public transport trip planners are electronic tools where travelers can request a travel plan for a given time and date and origin and destination. The usage data consists of these user requests in combination with the travel plans which are consulted by the user.

Previous research into trip planners (e.g. Brakewood & Watkins, 2018) shows that trip planners reduce the (perceived) waiting time of their users, reduce the (perceived) travel time of their users and increase the public transport demand. Brakewood & Watkins (2018) also show that these kinds of real time travel information systems influence different choices like travel, mode, route, boarding stop and departure time. We did not find research focused on forecasting ridership with trip planner usage data. However, trip planner usage data could provide valuable information on the (short-term) transport demand.

This explorative research investigates the predictability of ridership of public transport by using trip planner usage data. The log data of the trip planner are fused with the transaction data of smart cards to investigate the correlation between consulted trips and trips made.

We will utilize a case study for this research. The case study consists of the bus network in Groningen and Drenthe. The network is planned and scheduled by OV-bureau Groningen Drenthe and operated by Qbuzz. For this study the data of 9292 was used. 9292 is one of the major trip planners in the Netherlands and includes all public transport modes for the whole country. The 9292 data is fused with the transaction data of the OV-chipkaart, the Dutch smart card valid for all public transport modes across the country. This transaction data represents the realized demand.

If the ridership in a bus can be predicted, OV-bureau hopes to acquire new ways to improve their operations. For instance, through improving the planning of reinforcement buses. Currently, OV-bureau schedules most reinforcement buses a week in advance. The buses that are assigned a week in advance are scheduled to lines which were crowded the week before. This method is not very dynamic as it is based on the demand of last week and the expert judgement of the planners. With an appropriate forecasting model, OV-bureau could plan the reinforcement buses dynamically, e.g. only when insufficient capacity is forecasted. Preventing insufficient capacity would boost the public image of OV-bureau and public transport as a whole. For instance, there was an unexpected peak in passenger after the spring break in 2018 on lines with schools, which resulted in denied boarding and an article in the regional newspaper with the heading "Buses still on holidays, but pupils and students not" (Trimbach, 2018). In this case, OV-bureau was misinformed by one of the schools, but still held all the blame.

A second possible application of the new prediction method could be the planning of buses during large public events. Large public events or multiple smaller ones cause high variance in transport demand. As information on most events is limited and not centralized, their influence on the system is hard to predict (Pereira et al., 2015). The demand varies with the attractiveness of the event, the weather, whether the event is at night and whether people have to work the next morning. These buses are scheduled by OV-bureau based on trial and error and historical data. OV-bureau hopes to identify shortcoming supply before it happens and thus preventing inconvenience to passengers by forecasting the ridership. Unfortunately, the time period of the provided datasets does not include large events. Therefore, we cannot analyze the predictability of the demand in the scenario of a large event.

A third application is assigning the bus type dynamically. Changing to a smaller bus size could decrease the costs but also the carbon footprint.

Finally, the forecasts could be used to give more reliable information on the crowdedness in the bus. For instance, via the trip planner of 9292.

1.2 Research objective

The objective of this exploratory research is to determine whether usage data of a major trip planner can be used to predict the ridership of bus trips. There should be a correlation if the trip planner in question is widely used among the public transport user: If there are more travel advices consulted for a particular hour, it is likely there are more travelers intending to use public transport during that hour. However, for the forecast to be valuable to the transit operator, this correlation should have a certain accuracy for time and space. More specifically, the operator should be able to predict the ridership for a certain bus trip. Otherwise, the log data of a trip planner are not an effective new information source for forecasting shortcomings in bus transportation supply.

1.3 Research questions

In order to address the objective, the following main research question is formulated:

Can one forecast short-term ridership of buses using data containing the consulted travel advices from a widely used trip planner for public transport and what accuracy can one achieve in different scenarios?

By answering the following questions, the main research question will be answered.

1. *What internal and external factors cause fluctuations in bus transport demand according to literature?*

   This question helps to get a better understanding about varying public transport demand. Transport demand varies with time and space. But other internal factors like fares and the type of buses can also influence the demand. The result of this question is a list of factors which influences the transit demand in Groningen and Drenthe per bus line.

2. *What are the opportunities and challenges of using log data from the 9292-trip planner for forecasting ridership?*

   9292 is one of the major trip planners in the Netherlands and includes all public transport modes for the whole country. 9292 has designed algorithms to construct travel advices and a platform to communicate these advices with travelers. These designs effect which trips the traveler gets to choose from, which in turn effect the log data. This question investigates the consequences of selecting the log data of 9292 instead of other available trip planners. Furthermore, a list of requirements for trip planner data is derived.

3. *What are the opportunities and challenges of using OV-chipkaart transaction data to represent ridership?*

   The OV-chipkaart is the Dutch smart card valid for all public transport modes across the country. Travelers use the OV-chipkaart by tapping in and tapping out at the start and end of their journey and each time they change between operators or vehicles (except for trains and metros). However, this dataset is not all encompassing; there are still some other fare paying methods available. Therefore, it might be that some extra attention is needed when using data from a smart card. By answering this question, we create a better understanding on how to use the transaction data to represent ridership. We will also create a list of requirements for the smart card transaction data.

4. *To what extent does 9292 log data relate with ridership?*

By answering the last research question, we get a better understanding of the relation between 9292 and the ridership. With this understanding we can answer the main research question.

1.4 Outline

The remaining chapters in this report are as follows: In chapter 2 the literature will be reviewed to answer the question: *Which scenarios cause fluctuations in bus transport demand in Groningen and Drenthe?* Chapter 3 will discuss the used case study in three parts. In the fourth chapter the different data sources and used datasets will be introduced. The fifth chapter explains the used methodology. Chapter 6 will introduce the results. The discussion of these results can be found in chapter 7. The final chapter includes the conclusion to the research questions and implications and recommendations for practice and science.

To make the report more concise and easier to read, some parts are moved to the appendix. And the province Groningen will be abbreviated to just Groningen. And the city of Groningen will be referenced in full or as Groningen (city) to denote the difference between the two.

2 Literature review

2 Literature review

A literature review is conducted to answer research question 1: *What internal and external factors cause fluctuations in bus transport demand according to literature?* Internal factors are factors that can be regulated by the transit operator, like fare and frequency. External factors are all other factors.

The literature search is conducted as recommended by Webster & Watson (2002). The exact used methodology can be found in Appendix B. The found literature is presented in the concept matrix shown in Table 2.1. The concept matrix shows the forecast type, aggregation level and concepts (types of used factors) extracted from the respective article. A concept is only attributed to an article if the article actively included the concept.

Article	Forecast type	Spatial level	Temporal level	Temporal	Spatial/built environment	Demand characteristics	Weather	Event	Holidays	Transit characteristics	Other mode characteristics	Socio-economic	Socio-psychological
De Palma and Rochat, 1999	Survey	-	-				X				X		
Khattak and De Palma, 1997	Survey	-	-				X				X		
Chakrabarti, 2016	Mode choice	-	-							X	X	X	
Hensher and Rose, 2016	Mode choice	-	-							X	X	X	
Spears et al., 2013	Long term demand	-	-		X							X	X
Upchurch and Kuby, 2014	Long term demand	Station	Average weekday		X		X			X			
Brakewood et al., 2015	Long term demand	Route	Average weekday		X		X			X	X	X	
Kuby et al., 2004	Long term	Station	Average weekday		X		X			X		X	
Stopher, 1992	Long term demand	Route	Average time period	X	X					X		X	
Choi et al., 2012	Long term demand	Station to station (OD pair)	Average time period		X					X	X		
Doi and Allen, 1986	Medium term demand	Route	Month	X						X	X		
Tsai et al., 2009	Medium term demand	Station	Month	X					X				
Kalkstein et al., 2009	Short term demand	System	Day				X						
Guo et al., 2007	Short term demand	System	Day	X			X						
Li et al., 2014	Short term demand	Average route	Average day	X			X		X	X			
Jiang et al., 2014	Short term demand	Station	Day			X							

Article	Forecast type	Spatial level	Temporal level	Temporal	Spatial/built environment	Demand characteristics	Weather	Event	Holidays	Transit characteristics	Other mode characteristics	Socio-economic	Socio-psychological
Ni et al., 2017	Short term demand	Station	4 hours			X		X					
Van Oort et al., 2015a	Short term demand	Station	Morning peak			X				X			
Van Oort et al., 2015b	Short term demand	Station	Hour			X				X			
Zhou et al., 2017	Short term demand	System and station	Hour	X			X	X	X				
Pereira et al., 2015	Short term demand	Clustered stations	30 minutes	X	X			X					
Rodrigues et al., 2017	Short term demand	Station	30 minutes	X	X			X					
Xue et al., 2015	Short term demand	Route	15 minutes			X							
Li et al., 2017	Short term demand	Station	15 minutes			X							
Sun et al., 2015	Short term demand	Station	15 minutes			X							
Ding et al. 2016	Short term demand	Station	15 minutes	X	X	X				X			
Ohler et al., 2017	Short term demand	Vehicle	Stop passage	X		X	X	X	X	X			
Zhang et al., 2017	Real-time demand	Vehicle	Stop passage			X							

Table 2.1: Concept matrix

Existing researched transit demand forecasting models can be categorized based on the length of the prediction horizon. Long-term models - with a prediction horizon of a year - are mainly used to help decide on capital-intensive transit-oriented investments and to investigate the impact of major changes in service and environment. Short-term models – with a prediction horizon of days or hours - can be used by public transport operators to increase/decrease supply (dynamic traffic management) and to timely notify travelers on possible crowding (Pereira et al., 2015). Most models focus on predicting regular demand (Li et al., 2017). Beside the development of demand forecasting models, surveys are used to investigate the stated preference of travelers regarding varying factors (Khattak and De Palma, 1997; De Palma and Rochat, 1999) and to develop mode choice models (Chakrabarti, 2016; Hensher and Rose, 2016).

The demand forecasting models use different temporal and spatial aggregation levels depending on the factors they want to research. For instance, Pareira et al. (2015) used spatial aggregation when they summed the arrivals of stations and stops around large event venues to research the impact of events. Spatial levels vary from system wide to vehicle-stop passage level. Temporal levels vary from month to single stop passages.

There is a consensus in research in the influence of time and date. Research shows effects of seasonality, type of day and time of day. Xue et al. (2015) observe that the AM peak is sharper than the PM peak since schools and jobs start at the same time but end at different times. Some researchers cope with these fluctuations in demand by calibrating a model per time period. For instance, Stopher (1992), calibrated a separate model for peak (combination on AM and PM peak), day and night for weekdays and Li et al. (2014) developed a model per season. Others tried to forecast the impact of time by including dummy variables for type of day and time period (Ohler et al., 2017). Tsai et al. (2009) coped with seasonality in the data by using a moving average. Some tried to cope with these fluctuations by converting the variables to relative ones. For instance, Zhou et al. (2017) utilizes the ridership for a given hour and weekday compared to the monthly average for that hour and weekday. This way the intraday trends and patterns in ridership are accounted for.

Spatial features (synonym for attributes or variables) are also considered important. In some articles this feature is avoided. For example, by only forecasting the ridership on route-level for one route. Other models use spatial features like built environment to denote the attractiveness of a stop or route for travelers. For instance, Kuby et al. (2004) included variables indicating the intermodal connectivity of a station, such as accessibility, connecting services, park spaces, neighboring airports, type of station and variables as population and employment within walking distance.

Another used explanatory variable for ridership is historic demand. Xue et al. (2015) used lagged demand with a week, day and 15 minutes interval to forecast ridership. Li et al. (2017) also uses lagged demand augmented by lagged demand of 18 other major stations in the system to predict the number of passengers alighting major metro stations in Beijing during special events. Ding et al. (2016) uses passenger counts of nearby bus feeder services to predict short term ridership for the metro. Van Oort et al. (2015a) and Van Oort et al. (2015b) used historic demand to represent the demand in the base scenario.

The weather can also impact ridership. Kalkstein et al. (2009) show that air masses have an influence on ridership. This influence is stronger in the weekends as more trips are discretionary (Zhou et al.,2017). The effect is smaller during weekdays as these trips are mostly made by commuters which have to reach the destination. It is possible that the people that change from public transport to private mode to avoid long walking in bad weather cancel out the people that go from private to public transport to avoid congestion. Furthermore, Kalkstein et al. (2009) conclude that the effect of seasons is little. Ridership is more dependent upon relative weather conditions. A travel survey conducted by De Palma and Rochat (1999) in Geneva shows that around 40% of the commuters are influenced by adverse weather conditions in their travel choices. They report that departure time choice was more affected as mode and route choice. These results are similar to the travel survey conducted by Khattak and de Palma (1997) in Brussels. Li et al. (2014) observed a negative influence of humidity, wind speed, rainfall and temperature on bus ridership in a region in Shanghai. They used absolute values of the weather variables but note that this approach may introduce influences not solely based on weather but also on seasonality due to the presence of a weather pattern throughout the year. Guo et al. (2007) utilizes relative weather variables for this reason. In their research they also noted that the weather-ridership relationship is more complex because it is based on how individuals perceive and prepare for the weather, the presence of a lagged effect and because some weather variables correlate while others are synergetic. Li et al. (2014) also observe that there is no consensus on the specific influence of weather variables. For instance, some studies showed a positive correlation between temperature and ridership whereas other studies showed the opposite. As stated in their paper, it could be that some study areas have a higher active mode share than others, resulting in a modal shift to walking and cycling when the temperature rises. Beside the direct influence of weather on travelers, it also has an indirect influence on the journey. For instance, adverse weather lengthens running times, dwell times and disrupts service reliability (Guo et al., 2007).

Research agrees that events cause additional ridership. Kalkstein et al. (2009) stated that during events and festivals the ridership significantly changes. Rodrigues et al. (2017) tried to model this change utilizing information from the internet obtained via scraping and APIs on events to forecast the additional demand. Pareira et al. (2015) also scraped the information from the internet. They researched if the impact of an event could be predicted by including event information like event type/category, time to next event and a variable denoting if there is an event that day or not. Ni et al. (2017) used twitter as a social media source to identify the popularity of an event and used the number of tweets and unique users to forecast the passenger flow. Some studies try to avoid the influence of events and holidays and therefore select a timespan in which they don't occur (Zhou et al., 2017).

Holidays also impacts ridership. Doi and Allen (1986) observe less demand during the summer holiday. Ohler et al. (2017) included three different type of holidays: public holidays, school holidays and semester breaks of the local university. They researched two versions of representing these factors: via a binary dummy variable and via four dummy variables (days since the start, days left, days until next and days since previous). They propose the latter more elaborate way since they observe that demand also shifts just before and after a holiday. This demand shift was also observed by Kalkstein et al. (2009). To avoid these influences, they discarded these days from the dataset.

The characteristics of the public transport system also impact the ridership. Van Hagen (2011) adapted the pyramid of Maslow towards customer needs. The adapted pyramid consists of the layers (in order of importance); safety & reliability, speed, ease, comfort and experience. Li et al. (2014) use a cluster analysis to develop 3 clusters based on average headway, route length, number of bus stops, type of route (within district, urban-suburban and between districts) and crowdedness. Per cluster they developed different forecasting models. They observe that the influence of other variables is dependent on the bus route type. Brakewood et al. (2015) show that the introduction of real time travel information coincided with an increase in ridership. Stopher (1992) utilizes buses per hour, a measurement for the number miles driven in the service period and the time of one round trip. Kuby et al. (2004) incorporate station spacing. Upchurch and Kuby (2014) use a centrality measure to denote the average travel time to all other stations. On a larger scale they incorporate a variable denoting the urban area the total system coverages. Van Oort et al. (2015a) forecast the short-term demand using seat and crush capacity and Van Oort et al. (2015b) use other characteristics like the travel time.

Li et al. (2014) suggest that depending on the trip distance, external factors have a different level of impact. Guo et al. (2007) reasons that weather has an influence depending on infrastructure, trip characteristics, service characteristics and socio-economics. Travelers make certain travel decisions based on perceived comfort. This could depnd on the shelter at stops and stations, climate control systems in vehicles, headways, purpose and access to other modes.

Thus, the presence and status of alternative modes also play a role. De Palma and Rochat (1999) and Khattak and de Palma (1997) show that congestion leads to a modal shift from car to transit. Doi and Allen (1986) included gasoline prices and bridge toll prices in their forecasting model. And Chakrabarti (2016) used the travel time by transit compared to the travel time by car as input variable.

Socioeconomic variables are also widely used to explain transport flows and mode choices: Spears et al. (2013) and Chakrabarti (2016) use the number of cars per household as an input variable and Li et al. (2014) recognize the impact of socioeconomics features on total ridership. They kept their dataset within a year to limit the impact of a change in these factors.

Spears et al. (2013) also utilizes sociopsychological factors to explain ridership. Amongst these factors are attitudes towards transit and perceived safety.

The above described factors influence the public transport demand. More specifically they influence the travel choices of (potential) travelers directly, which in turn cause a change in demand. Current travel behavior research uses utility theory to explain travel choices, like whether to travel or not, destination choice, mode choice, route choice and departure time choice. Of these, route choice and departure time choice are short-term and thus sensitive to situational factors. These travel behavior models calculate utility costs based on some aspects of the journey like costs, time (acces, egress, travel or waiting) and comfort (De Donnea, 1972). Furthermore, the travel behavior also differs depending on the trip purpose and sociodemographic factors. For instance, commuters value time higher as noncommuters who are more sensitive to costs. Also, the availability of a car influences the mode choice. But even if a household is in possession of a car, it might not be available because another family member is using it or the person has no valid driver's license (Chakrabarti, 2016). Spears, Houston and Boarnet (2013) summarized the factors affecting transit use in the group's physical environment and cognitive processes (see Figure 2.1). The physical environment directly affects the behavior. Cognitive processes summarize the attitudes, social and personal norms, perceived control and habits of the individual. The factors in these two groups lead to the current travel behavior which result in a person-environment fit and certain short term and long-term adaptive actions, like changing the departure time or moving houses.

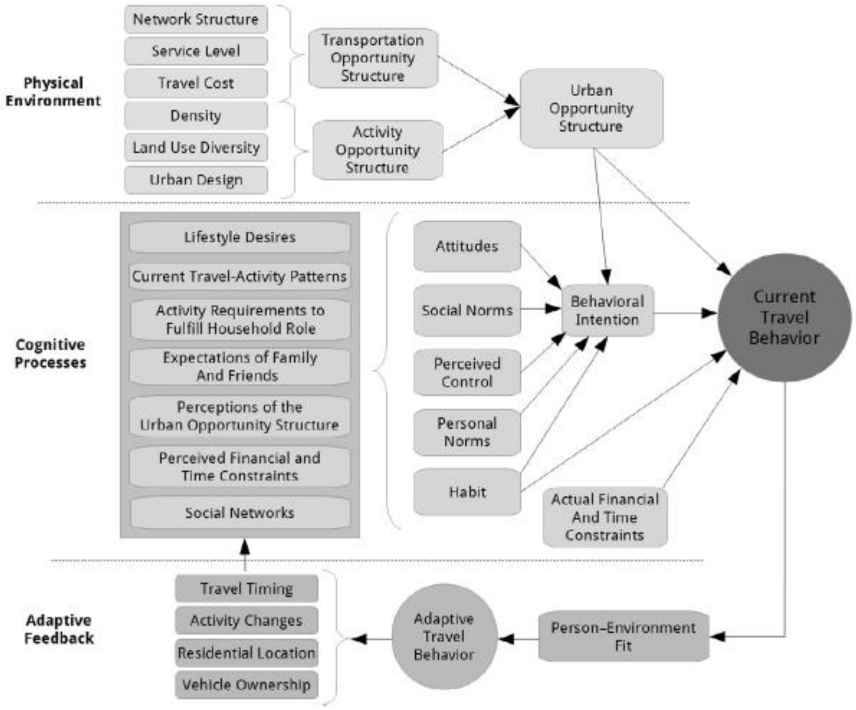


Figure 2.1: Framework of factors affecting transit use. Adapted from (Spears, Houston and Boarnet, 2013).

2.1 Conclusion

The variables used in the different studies differ a lot. Depending on the time, location and level of temporal- and spatial aggregation, the impact of variables differs. The used spatial and temporal level is generally chosen so that the input variables fluctuate. Long term demand forecasting models use variables that only change slowly over time. Medium term demand forecasting models use variables that change per month or season. Short term demand forecasting models use variables that change with the time unit used in the forecasting method, such as lagged demand, occurrence of events and weather.

The different variables can be categorized in the following groups: Temporal, Demand characteristics, Weather, Event, Holidays, Transit characteristics, Other mode characteristics, Spatial/built environment, Socio-economic and Socio-psychological. The first six of these groups can be useful to predict short term demand. Variables form the last four groups vary mostly only on the long-term. Depending on the location, time and aggregation level different variables are used.

Even when the influencing factors are known it matters how they are used as input in the model. For instance, it is possible to use relative values, moving averages or it could be useful to divide the variable in multiple dummy variables.

3 Case study

859
590
431
561
748
685
314
16
278
708
946
249
664
107
350
175
652
972
284
600
463
41
214
734
394
708
893
956
986
291
601
60
979
352
4
701
874
438

3 Case Study

For this master thesis we will utilize a case study in order to answer the main research question. In this section we will describe this case study and its scope. First, we will describe the spatial setting, followed by the public transport network, and finally we will discuss the temporal setting.

3.1 Region

The case study consists of the provinces Groningen and Drenthe, which are in the North-East of the Netherlands. To get a better understanding of the line network and the demand characteristics and to make it comparable to other researches in other cities, regions and countries, we will discuss some general statistics for these two provinces.

Groningen has a land area of 2,333 km^2 and Drenthe has a land area of 2,639 km^2 (see Appendix I for a map). These provinces consist of 68 municipalities of which Groningen is the largest and most well-known. Around 1 million habitants lived in these provinces in 2016, 490 thousand in Drenthe and 580 thousand in Groningen. About 19 percent of the habitants lived in the municipality of Groningen. The four biggest municipalities are in order of number of habitants: Groningen, Emmen, Assen and Hoogeveen. These four municipalities cover around 40 percent of the habitants (Central Bureau for Statistics, 2018a).

Figure 3.1 left shows the number of habitants per postcode-4 area[1]. The right side of Figure 3.1, shows the postcode-4 areas with over 5000 habitants, as can be seen these areas are limited and clustered around a few corridors. Thus, most areas have less than 5000 habitants.

[1] The postcode (postal code) system is used in the Netherlands to indicate groups of addresses. An intact postal code exists of 4 numbers and 2 letters and points to a (part of a) street. Without the 2 letters, we get the postcode-4 area, which indicates a part of a city or town.

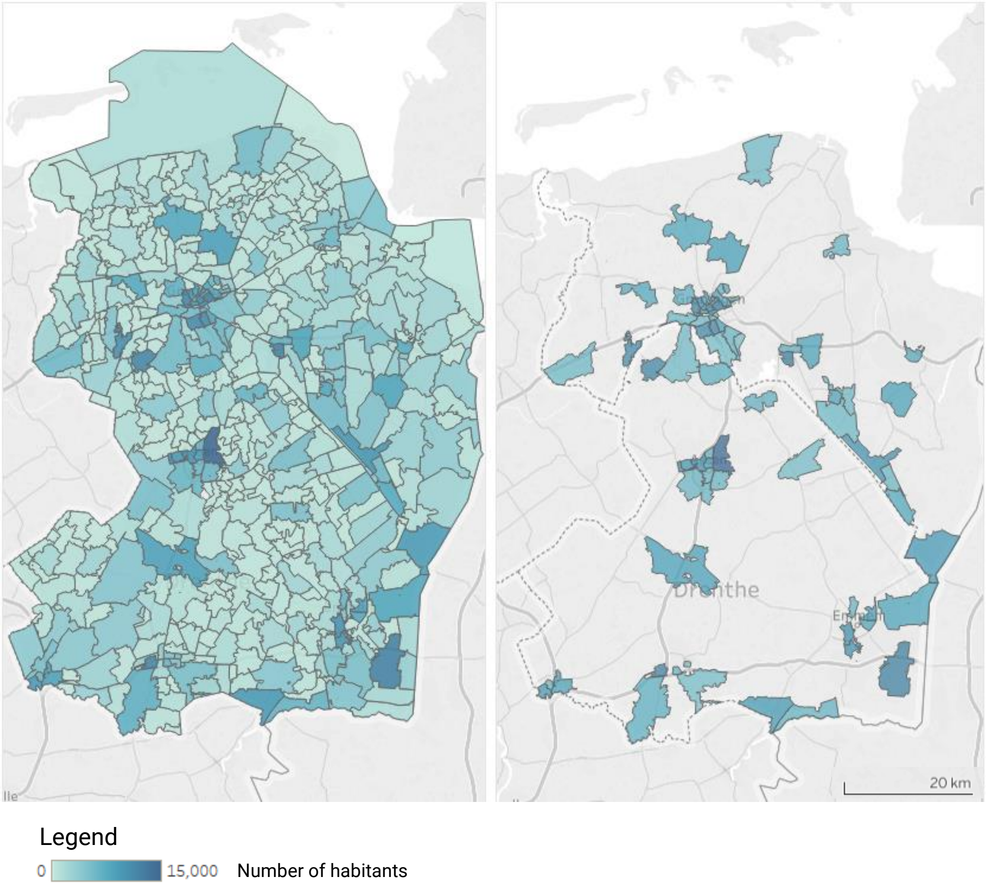


Figure 3.1 Left: Number of habitants per postcode-4; right: the postcode-4 areas with over 5000 habitants (Central Bureau for Statistics, 2017a)

Figure 3.2 shows the rate of habitants per urbanization category for the Netherlands and the provinces Groningen and Drenthe. Following this figure, Drenthe has a different distribution of urbanization than average in the Netherlands. Where on the national level the rates for high and strong urbanization are slightly higher as the rest, the rates in Drenthe show a steap curve with almost no highly urbanized areas and a clear peak of areas with no urbanization. Groningen shows a similar trend to Drenthe, but the trend is less distinct and Groningen has a peak at high urbanization, caused by the city of Groningen. From this picture we can conclude that in Groningen and Drenthe a large portion of people are living in less urbanized areas.

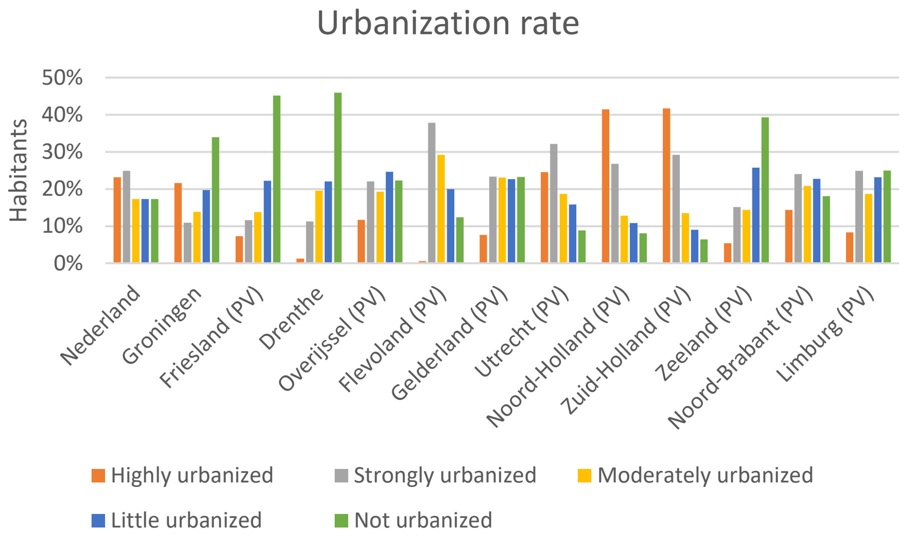


Figure 3.2 Rate of habitants per urbanization category in the Netherlands and the provinces Groningen and Drenthe using the average area addresses density (AAD) as measure (Central Bureau for Statistics, 2018b): Highly urbanized – AAD of 2,500 or more addresses per km²; Strongly urbanized - AAD between 1,500 and 2,500 addresses per km²; Moderately urbanized - AAD between 1,000 and 1,600 addresses per km²; Little urbanized - AAD between 500 and 1,000 addresses per km²; Not urbanized - AAD of less than 500 addresses per km².

3.2 Public transport network

In the case study we will use data from all the bus lines operated by the bus operator Qbuzz in Groningen and Drenthe. In this section these lines and the overall network is discussed. This will give a better overview of the network and will help understand certain travel behavior (demand pattern and trip planner usage) caused by the network characteristics. For instance, a frequent service (every 10 minutes a bus) connecting two stops results in less need for making a pre-trip plan by travelers and a faster connection with fewer transfers (e.g. more comfort and less transfer/waiting times) and therefore is more attractive for potential travelers. Thus, it is important to understand with what kind of network we are dealing with.

The basic dilemma of constructing a public transport network is to find a balance between travel times and operation and investment. Travelers value a short travel time. In a full connected network, a network where all stops are directly connected by a bus, the travel time is the shortest. However, the operational costs involved for such a network are high: It would either require a high (expensive) capacity to ensure acceptable frequencies or the frequencies would be low resulting in long waiting time. An optimal network for the operator would be one with a minimal spanning tree, but this would result in larger travel times. Thus, the goal is to connect the stops optimally, resulting in minimal waiting and in-vehicle time, given financial and operational constraints. Egeter (1993) summarized this in four design dilemmas':

1. Stop density (the number of stops per square kilometer): A network with a high stop density results in a lower access and egress time. However, more stops result in more stopovers for buses and thus longer in-vehicle times.
2. Network density (the total length of used links per square kilometer): A network which is more connected results in lower in-vehicle times. However, the same number of buses have to be divided over more links, thus the network is less frequent, and the waiting time increases.
3. Line density (the total length of lines per square kilometer): A network with a higher line density result in fewer transfers. However, the frequency per line will be lower which result in higher waiting times.
4. Number of network levels, (e.g. national, regional, urban, etc.): Multiple network levels result in lower travel time as each network level can serve a specific trip length best. However, by introducing more network levels, you also introduce transfers.

It is also important to note that the network design is limited by the existing spatial structures in cities and regions. Therefore, line spacing is limited by the road spacing. And special buildings like the hospital and university, might influence the network.

3.2.1 Network levels

In this section we will explore the current public transport network in Groningen and Drenthe. The map of the public transport network is given in Figure I.1 (Appendix I) and Figure 3.3.

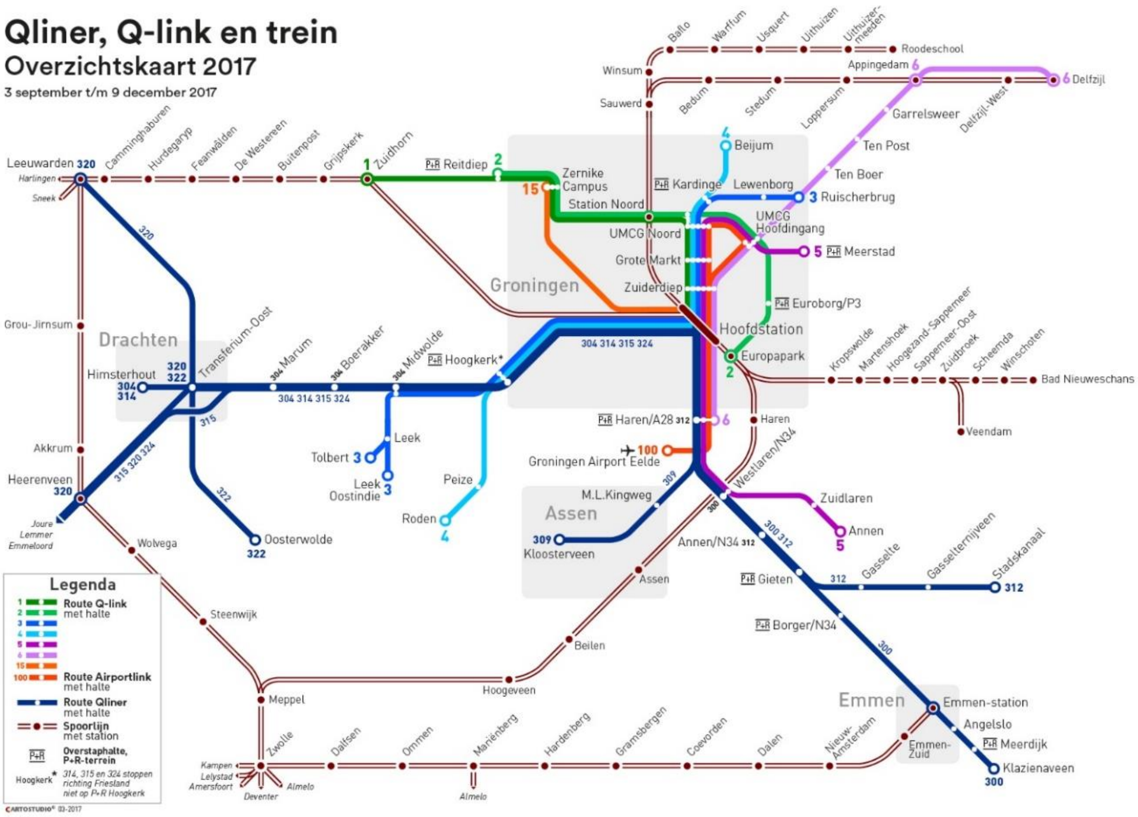


Figure 3.3: Network map of the train, Qliner and Q-link. Retrieved from https://qbuzz.nl/GD/onderweg/waarmee-reis-ik/qliner/

The public transport network in Groningen and Drenthe has multiple network levels. The NS (the biggest Dutch railways operator) operates trains nationally. These trains are called intercitys. Intercitys connect the major cities directly. NS operates one intercity line in Groningen and Drenthe: From the city Groningen directly to Assen and continuing south west. Along the way these intercitys also stop at transfer stations, which make it possible to transfer to intercitys and regional trains to other parts in the country. Because of this direct service, the stations of Assen and Groningen act as the main access/egress points for public transit users exiting and accessing the provinces of Groningen and Drenthe.

NS also operates regional trains which serve the smaller stations on the same corridor as the intercitys. In addition, Arriva (railway and bus operator) operates a regional train service between Leeuwarden and Groningen, two services from Groningen to the north, one from Groningen to the south East and a regional service between Emmen and Zwolle.

The regional bus service operates on the same level as the regional train service. The regional buses are operated between villages and towns in Groningen and Drenthe. On workdays the frequency is one or two buses per hour. In the evening and in the weekends the lines are operated less frequent. Some lines are operated off-peak hours by a LijnBelBus (literally: line-phone-bus, a smaller bus which you have to book by calling them) (Qbuzz, 2018).

The regional level can be divided further in the Qlink and Qliner, see Figure 3.3. Qlink exists of 7 lines. 6 Lines are between the bigger living- and workplaces in the region and Groningen. And one line is between Groningen central station and Zernike (a major workplace). These lines have a higher operating speed, because of less stops and some dedicated lanes, have a high operating frequency - during peak period every 10 minutes or more often - and are more luxurious (Qbuzz, 2018).

The Qliner operates fast direct routes between the bigger cities and villages and Groningen. The routes are thus longer than the Qlink routes, but other than that the Qliner is similar to the Qlink (Qbuzz, 2018). As you can see in Figure 3.3, the line network of the Q-link, Qliner and train has a star shape with Groningen in the middle.

One network level further are the city buses. In Groningen, Assen, Emmen, Hoogeveen, Meppel and Veendam lines are operated on a city level. These lines stop often and have a frequency of two buses (or more) per hour (Qbuzz, 2018). The last network level contains the buurtbus (english: neighbourhood-bus) which is organised locally and is operated by volunteers. Other options, like FlixBus, are left out of the scope.

3.3 Time period

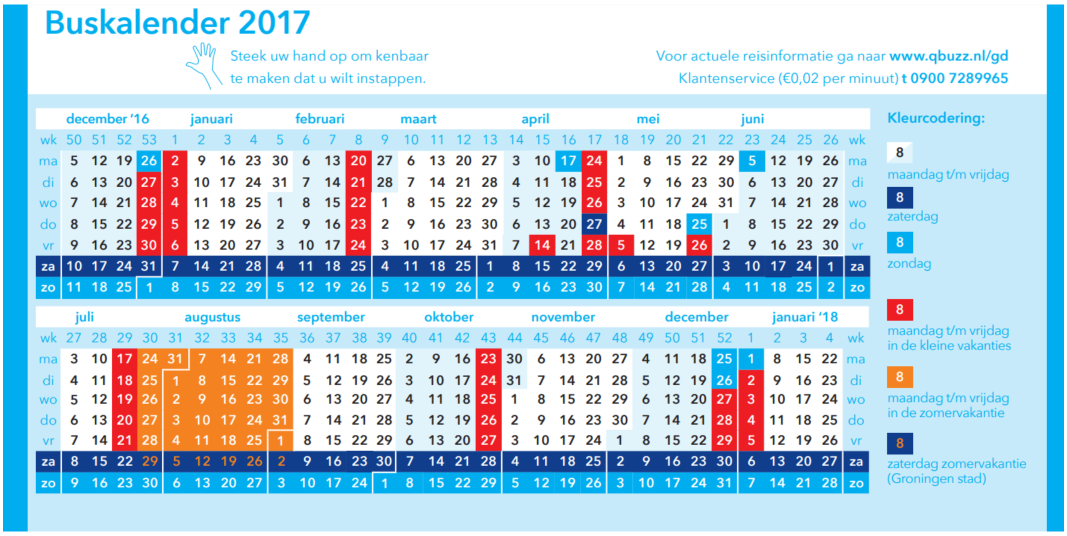
For the case study we will use data from the first few months of 2017 between January 1st and March 31st. This was the most recent data available at the time. The last change in bus schedule was on December 11th, 2016. During this period of time there was no extreme weather, strikes or other significant disturbances for daily operations.

Time and day have a significant influence on the demand and type of traveler, as was shown in the literature study. On weekdays there are relatively more commuters whereas in the weekend, during holidays and in the evenings relatively more trips are made by travelers with recreational objectives. To adjust the supply as much as possible to the demand OV-bureau works with 6 types of days:

1. Weekdays (Monday till Friday)
2. Saturdays
3. Sundays and national holidays
4. Weekdays during small holidays

5. Weekdays during the summer holidays
6. Saturdays during the summer holidays (for Groningen city)

See Figure 3.4 for the annual planning of the operational day schedules. In our research period the Saturdays and Sundays have their corresponding day schedule. There are two small holidays: the first week of the dataset (which started a week earlier) and between Monday 20 February and Friday 24 February. The rest of the weekdays are scheduled as ordinary weekday.



Buskalender 2017

Steek uw hand op om kenbaar
te maken dat u wilt instappen.

Voor actuele reisinformatie ga naar **www.qbuzz.nl/gd**
Klantenservice (€0,02 per minuut) **t 0900 7289965**

	december '16					januari				februari				maart					april				mei				juni				
wk	50	51	52	53	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	wk
ma	5	12	19	26	2	9	16	23	30	6	13	20	27	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	ma
di	6	13	20	27	3	10	17	24	31	7	14	21	28	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	di
wo	7	14	21	28	4	11	18	25	1	8	15	22	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	wo
do	8	15	22	29	5	12	19	26	2	9	16	23	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	do
vr	9	16	23	30	6	13	20	27	3	10	17	24	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	vr
za	10	17	24	31	7	14	21	28	4	11	18	25	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	za
zo	11	18	25	1	8	15	22	29	5	12	19	26	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	2	zo

		juli				augustus					september				oktober					november				december				januari '18			
wk	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	1	2	3	4	wk
ma	3	10	17	24	31	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	ma
di	4	11	18	25	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	di
wo	5	12	19	26	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	wo
do	6	13	20	27	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	do
vr	7	14	21	28	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	vr
za	8	15	22	29	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	za
zo	9	16	23	30	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	zo

Kleurcodering:

8 maandag t/m vrijdag

8 zaterdag

8 zondag

8 maandag t/m vrijdag in de kleine vakanties

8 maandag t/m vrijdag in de zomervakantie

8 zaterdag zomervakantie (Groningen stad)

Figure 3.4 The bus schedule for 2017. Image adapted from https://qbuzz.nl/GD/files/3414/8007/6387/Buskalender_2017_def_v5.pdf accessed 26-06-2017

All days in a category have the same planned day schedule. However, it could be that because of some roadworks or because of extra demand due to a public event bus routes are changed or additional buses are used. These changes are known beforehand and can therefore be anticipated. The used time period does not contain major events which require extra buses. There are two events for which some extra buses are planned: the open house of the University of Groningen on Friday February the 3rd and the Monnikenloop on Saturday March 25th. Thus, we cannot investigate the usefulness of trip planner usage data for large events.

3.4 Conclusion

In this chapter we introduced the case study we will utilize. We have data available for the first three months of 2017 for the bus lines operated by Qbuzz in Groningen and Drenthe. The region is suitable to analyze effects on lines running through low density areas as well as high density areas. The data of Qbuzz allows us to analyze effects for urban and regional lines. OV-bureau utilizes standard schedule days with minimal variation in between the schedule day. This allows us to analyze the same bus trip in different temporal settings.

4 Data preparation

4 Data preparation

We will predict the ridership of buses in the provinces Groningen and Drenthe using the usage data of 9292 (a major trip planner for public transport in the Netherlands) and the transaction data from the OV-chipkaart (the Dutch smart card which is valid for all public transport in the Netherlands). In this chapter we will describe the context of these datasets. This will help to make sense of trends and artefacts in the datasets.

To forecast the ridership of buses we need data on the number of people boarding and alighting a stop and the number of people that got the advice to board and alight a bus at that stop, see Table 4.1 for an example. More specifically, we want this data at the vehicle-stop level.

Date	Bus line	Bus trip	Bus stop	Stop order	Number of people boarding	Number of people alighting	Number of people boarding according to 9292	Number of people alighting according to 9292
01-01-2017	g554	1002	Roden, Dorth	1	4	0	6	0
01-01-2017	g554	1002	Roden, Kastelenlaan	2	1	0	0	3

Table 4.1: An example of the desired final dataset at the vehicle-stop level.

We will gather and construct these data by merging four datasets. One of the datasets contains the travel advices which were consulted by users of the 9292 trip planner application, hereafter this dataset will be called Trip planner data. The second dataset contains transaction data of the OV-chipkaart, hereafter this dataset will be called Smart card data. These two datasets contain the records on origin-destination level; e.g. boarding the bus X at stop A at time Y and alighting the bus at stop B at time Z. Unfortunately, for both these datasets there is no direct relation stored with the vehicles or bus trips (bus X is unknown). We have to preprocess the data to discover the used bus and convert the dataset to the vehicle-stop level. An intermediate step relating the trip planner data and the smart card data to bus trips is needed. We will use an extra dataset to do so. This extra dataset, hereafter called bus data, contains information on the position and time of buses, the timetable and the delays. The last dataset contains data on the weather which will be used to investigate the influence of the weather. This last dataset will be included since the literature review suggested the existence of a correlation between weather and public transport demand.

The first three datasets are somewhat related to each other, since they use the data from the same organization: *NDOV (National Databank Public Transport)*. Information about the bus timetable and the current bus status are used by different organizations. These data and other related information are collected by NDOV and are publicly made available via different two portals. One of the portals is maintained by 9292 and is accessible via https://www.reisinformatiegroep.nl/ndovloket/. These data are made available via different interfaces (*koppelvlakken*). For instance, *koppelvlak 1* (KV1) contains the timetable. Data of KV1 do not change much over time. *Koppelvlak 6* (KV6) is used for sending information during the bus trip about the execution of this bus trip. There are constantly messages coming in directly from the buses via this KV. Furthermore, via other *koppelvlakken* of NDOV, operators are able to communicate with dynamic displays which are present at some stops to inform the travelers.

9292 is one of the users of these *koppelvlakken* (interfaces). To access the timetable, to account for any current delays and to get information on the fare. Figure 4.1 shows a scheme of the information flow.

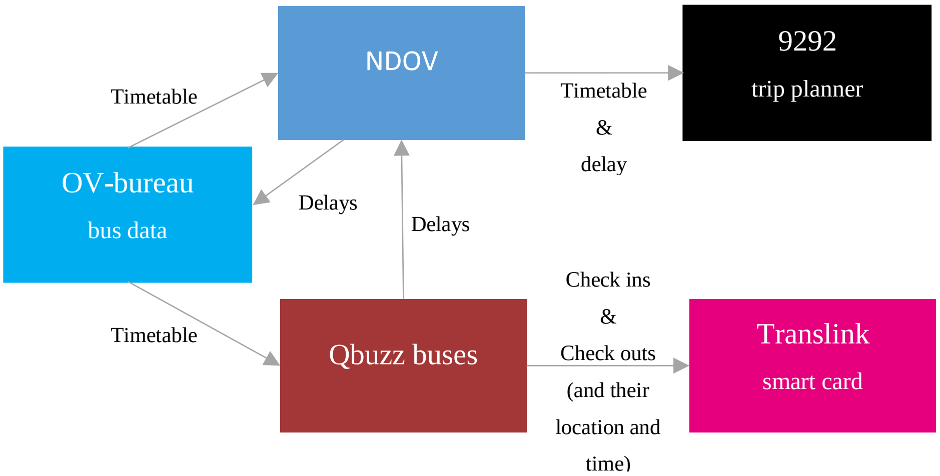


Figure 4.1: The information streams between the different datasets.

The outline of this chapter is as follows; First we will discuss how we will handle noise in section 4.1. Sections 4.2 to 4.5 discuss the four datasets in 4 steps; 1: What entails the dataset, 2: How is the data collect, 3: How is the data preprocessed and 4: What trends are visible in the data. In section 4.6 we discuss how we merged the datasets. Section 4.7 highlights trends in the combined dataset and in section 4.8 the features are discussed.

4.1 Handling noise

Before we elaborate on the four datasets and the merging of these datasets, we will discuss how we will handle the encountered noise. Noise handling is an important step during data preprocessing.

Van Der Spoel et al. (2012) differentiates between 3 types of noise: *sequence noise* (noise in the order of events), *duration noise* (missing or wrong timestamps) and *human noise* (noise due to human error). It is not likely to encounter sequence noise during this study since the order of events is stated by the bus timetable. However, it is likely that both duration noise and human noise occur.

Teng (1999) enumerates three methods of handling noise. The first method is keeping the noise to prevent the predictive model from overfitting. The second method is to discard the noise beforehand. The third method is to find the noise and try and correct it. We will first try to find and correct the noise and if it turns out to be impossible, we will discard the data.

4.2 Dataset 1 - Bus data

The first dataset we will discuss is the bus data dataset. We will use this dataset for the public transport supply. The dataset is provided by OV-bureau which maintains a database with data extracted from NDOV.

4.2.1 Description

The bus data dataset contains detailed information about all the trips on the vehicle-stop level. This information includes the route, the stop order, the planned time of arrival and departure, the current delay, if the bus was cancelled, etc. Thus, this dataset contains valuable information on the timetable and the execution of this timetable.

OV-bureau has contracted two operators: Qbuzz and Arriva Touring. For this case study we will only use the data of Qbuzz, since they operate on the major part of the network and only the smart card data for this operator are available.

4.2.2 Data collection

Initially the dataset is collected by OV-bureau. Both k*oppelvlak 1* and an abstract of *koppelvlak 6* are used for this dataset. By merging these two *koppelvlakken*, OV-bureau constructed a database with a record for each time a bus is supposed to pass a stop (target arrival and departure time) augmented with information on the actual passage (recorded target and arrival time and recorded punctuality).

The data were provided by OV-bureau by means of a flat table. This flat table contains 17,094,510 records which represent all the bus passages of stops between 12 December 2016 and 17 May 2017 for the concession GD (Groningen Drenthe).

Table 4.2 lists the fields and their description as obtained from OV-bureau. An example of this data can be found in Appendix C. This dataset has to be preprocessed in order to make it suitable for matching. For further processing, the dataset was loaded into a SQL table.

Variable	Type	Description
concessieareacode	Text	The area code for the concession. In this dataset this code is ‘GD’.
dataownercode	Text	The owner of the data. Thus, for this dataset ‘QBUZZ’.
operationdate	Date, dd-mm-yyyy	The date
linepublicnumber	Number	The line number for public use.
lineplanningnumber	Text	The line number for internal use. The prefixes have a meaning, for instance *t* denotes an additional line for an event, *g* denotes a line in Groningen and *d* a line in Drenthe
lijnnaam	Text	Description of the line
tripnumber	Number	The trip number: Even numbers in one direction, uneven numbers in the other.
vehicleregistrationnumber	Number	Identifier for the used vehicle
userstopcode	Number	Identifier code for the stop
timingpointname	Text	Description of the stop
haltetype	Text	Indication if the stop is serviced at the begin, end or during the trip.
tijdhalte	Boolean	Indication if a stop is used for syncing with the schedule if needed.
userstopordernumber	Number	The sequence in which the stops are serviced during a trip.

Variable	Type	Description
targetarrivaltime	Timestamp without date	The target arrival time
targetdeparturetime	Timestamp without date	The target departure time
recordedDepartureTime	Timestamp without date	The actual departure time
RecordedArrivalTime	Timestamp without date	The actual arrival time
RecordedPunctuality	Number	The number of seconds the bus is delayed.
HasPassed	Boolean	If the bus has passed the bus stop.
HasStopped	Boolean	If the bus has stopped for boarding and alighting at the bus stop.
TripCancelled	Boolean	If the trip is cancelled, this can be done pre-trip or on trip. It is also that only one or a few stops are cancelled due to road works.
TripDispatched	Boolean	If *TripDispatched* is true the bus has passed this stop, however due to connectivity problems this was not logged. The remainder of the stops will then also have this variable set to true.

Table 4.2: The fields of the bus data set as provided by OV-bureau

4.2.3 Data preprocessing

A few steps are needed before this dataset is ready for usage; First we rectify the noise in the data where possible. Next we trim the data in order to fit the study area and the time span. We conclude by augmenting the dataset with useful features

A step to make the dataset easier to use is the augmenting of the timestamps; The timestamps of the target arrival and departure time and recorded arrival and departure time are lacking a date. The date present in the field '*operationdate*' can be used. However, an operation day is defined from 04:00 to 04:00 the next day. Since the dates in the other datasets are regular calendar dates, the operation dates have to be converted to calendar dates before they can be used to append the target and recorded times. For most records this is rather easy; If a target time or a recorded time is between 00:00 and 04:00, the operation date plus 1 day should be added to the time, otherwise just the operation date is sufficient. Four bus lines start before 04:00 and end after 04:00. For these bus lines all the datetimes of the records are increased with 1 day. These four bus lines are: 402 - Groningen - Vries [Nachtbus], 417 - Groningen - Roden - Leek [Nachtbus], 418 - Gieten - Groningen [Nachtbus] and 419 - Assen - Groningen [Nachtbus].

Also, the denotation of no recording for a departure time or arrival time was ambiguous. If there was no record these times were saved as '00:00:00' instead of a null value. Therefore, an extra step was needed to detect the records where '00:00:00' was the actual recorded time.

Furthermore, the field *'recordedpunctuality'*, which represents the observed delay, is not reliable. Most often this variable describes the number of seconds between the planned departure time and the real departure time. However, sometimes when the real departure time was missing, the recorded punctuality was based on the real arrival time. There are also lots of instances where there is a recorded punctuality, but no real arrival or departure time. For those instances the recorded departure time is calculated using the recorded punctuality and the target departure time.

There is noise in the data. The found anomalies are listed in Appendix D. Most anomalies are infrequent. However, it can be determined that the fields, especially those which are extracted from *Koppelvlak 6*, are not 100% reliable. These fields give feedback on how the bus executed the bus trip and are sent by the buses while on trip. It is not possible to precisely determine when the fields are erroneous. The occasions which are listed in the table stand out, because of the extremeness of the error. But less extreme errors are impossible to find easily and even when found it is unknown which field is erroneous. The biggest errors are rectified, but for the rest the dataset is used as is, while keeping in mind the possible errors in the data.

The data has to be trimmed in order to represent the same region and time as the smart card dataset and the trip planner dataset. The trimmed dataset contains 11,447,562 records between 01-01-2017 and 14-04-2017. We delete the records for stop passages outside Groningen and Drenthe (11 Bus lines traverse these borders, see Appendix E) and the records for stop passages before 23:00 on 31-12-2016 and after 01:00 on 15-04-2017 (we use this 1 hour extra of data to compensate for delays).

The dataset is augmented with the travel time since the last stop, the departure time of the last stop and the departure time of the next stop. The travel time since the last stop is used for exploratory data analysis. The departure time from the previous stop and the departure time at the next stop are used for the trip matching step as discussed later. The departure time is used since this is more frequently logged than the arrival time. If there is no recorded departure time available at the previous or next stop the target departure time is taken. In case the stop has no previous or next stop available the own recorded departure time was used plus or minus 30 minutes.

Moreover, the different datasets use different identifiers and aggregation levels for the stops. 9292 has defined the stops at one aggregation level higher in clusters, where stops with the same name are in the same cluster. For this analysis we will use the aggregation level used by 9292. Therefore, the records are also augmented with the 9292 stop clusters. The step where the bus data stops are matched to the 9292 clusters is described in Appendix F.

4.2.4 Data exploration

In this section we will highlight some key characteristics for this dataset.

Different factors influence the punctuality of the bus. Therefore, the recorded passage times often differ from the planned times. These recorded times are thus variable and continuous. Figure 4.2 is a histogram which shows the distribution of the recorded delay for each stop passage in the dataset. The distribution has a mean of 60.06 seconds and resembles a normal distribution with a positive skewness. However, the histogram also shows that more as 20% of the buses depart too early. Figure 4.3 show that the distribution of the recorded delay varies with time. Moreover, the distribution also has a larger spread during peak hours and during night time.

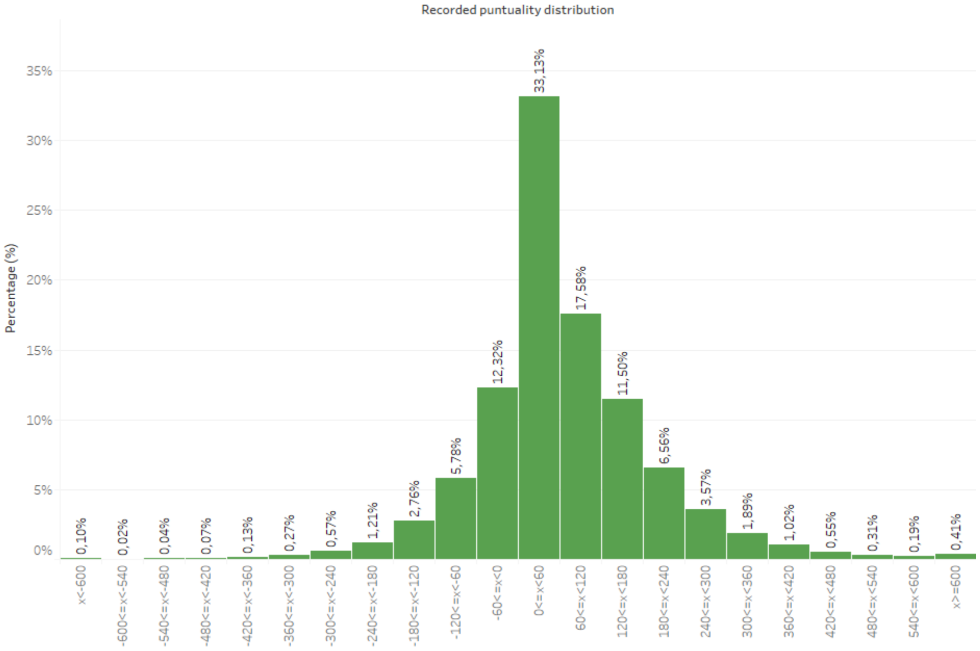


Figure 4.2 Histogram of the recorded punctuality distribution for all recorded stop passages. X= punctuality, in seconds

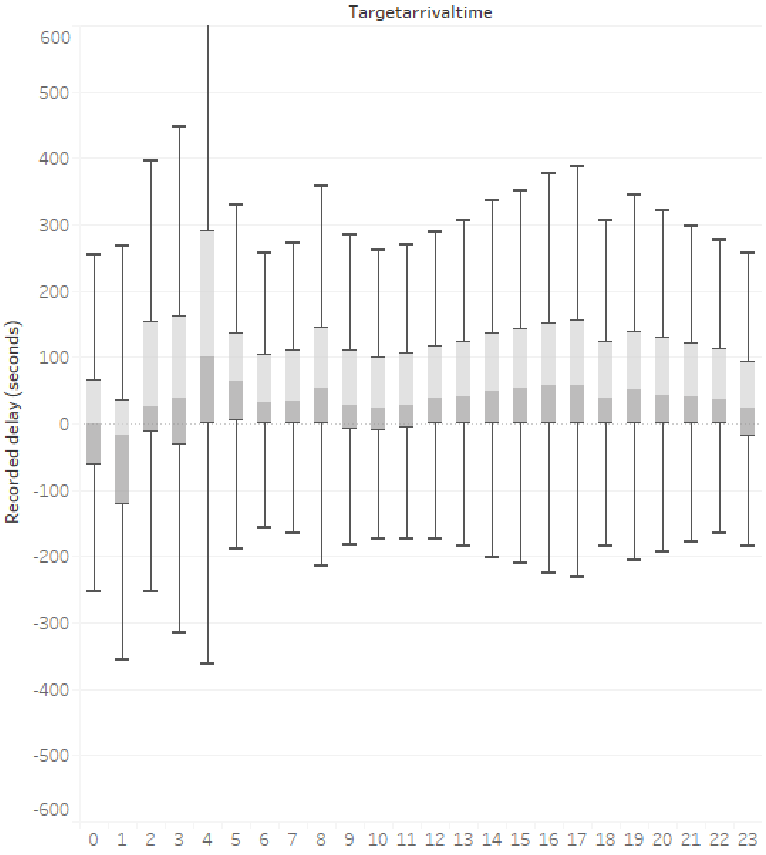


Figure 4.3 Distribution of the recorded delay aggregated over hours

Appendix G contains a table with all the variants of the routes available in the bus dataset. In this table, the line characteristics like average stop distance, travel time and number of trips recorded are shown. These lines are all the different kind of lines as described in section 3.2.1. However, some lines do not have AVL data available or do not have a smart card reader device, for example lines which contain a belbus. These lines are later ignored. Unfortunately, most lines have a few trips that are serviced by a belbus, for instance in the off-peak period. This might provide problems since these trips provide noise to the dataset. Figure 4.4 shows an example of how the variants of a line are constructed: a variant is unique in the stops, stop order (implicitly also includes direction) and the line planning number.

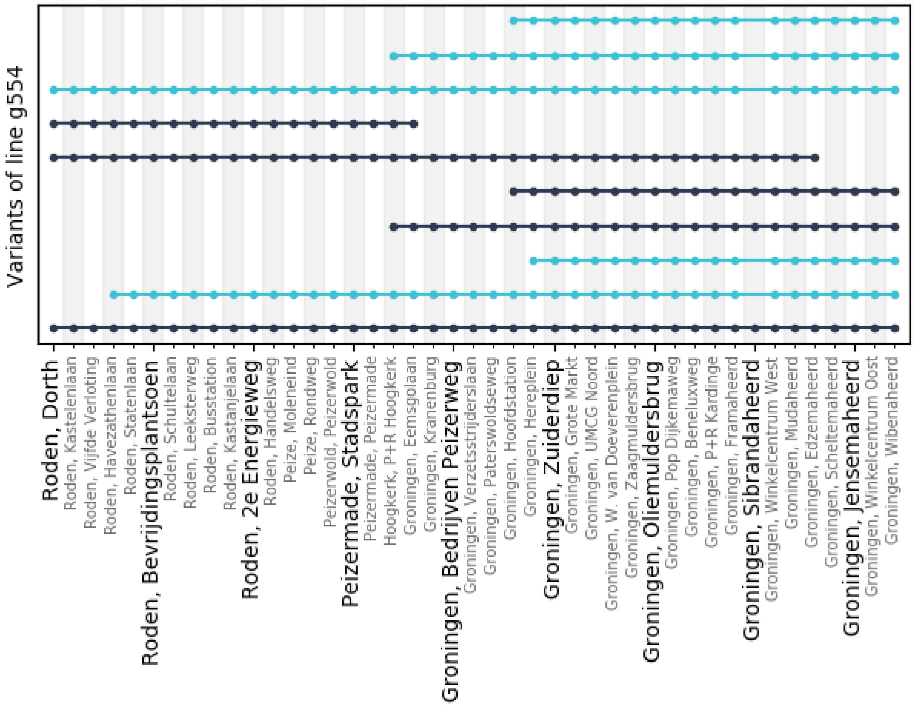


Figure 4.4: Visualization for the different variants of line g554. The color of the line represents the direction and a node represents that the bus does service that stops. Some variants are longer as others.

Figure 4.5 shows the distribution of the headway between two buses on the same route on stop level. The headway is defined as the time between two trips of the same line visiting the same stop in the same direction (having the same previous stop and the same succeeding stop) at the same operation date. The line on zero and the large peaks at certain intervals show that the timetable has a clear time pattern. Since some trips don't have a headway (the first trips of a day don't have any preceding trips) we will bin the headways. The used bins are shown by the shaded areas. It is assumed that a difference between a headway of 10 minutes or 15 minutes has more impact as a difference between 70 and 75 minutes. Therefore, the initial bins are smaller as later bins. Furthermore, each peak has its own bin. The last bin (> 125 min) also contains the records without a headway, since a headway of over 125 is similar to the characteristics of the first trip.

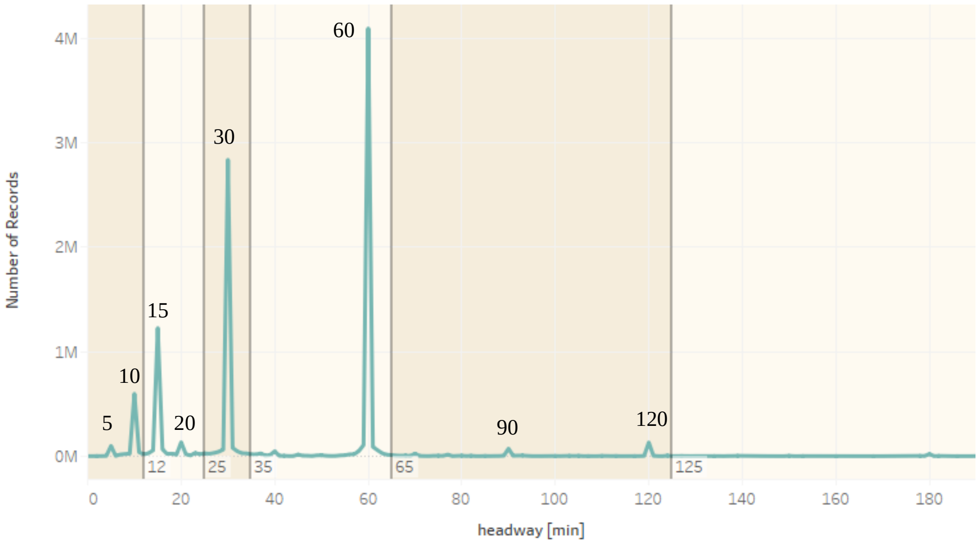


Figure 4.5: The distribution of the headway.

4.3 Dataset 2 - Trip planner

We acquired a dataset from 9292 to use as trip planner usage data. The dataset contains the travel information consulted by users in their trip planner.

9292 is not the only travel planner in the Netherlands used for public transport freely available to the public. Among the biggest travel planners are 9292, NS, ANWB, Google Maps and Go About. The travel planners differ in looks, included modes which are included and functionality. Users choose a travel planner based on these differences and their personal preferences. For this case study we utilize data from the 9292-trip planner because is widely used and well known. It is specialized for public transport since 1992 and the data is available to us. We will therefore only discuss how users use the 9292-travel planner to plan their journey.

4.3.1 Description

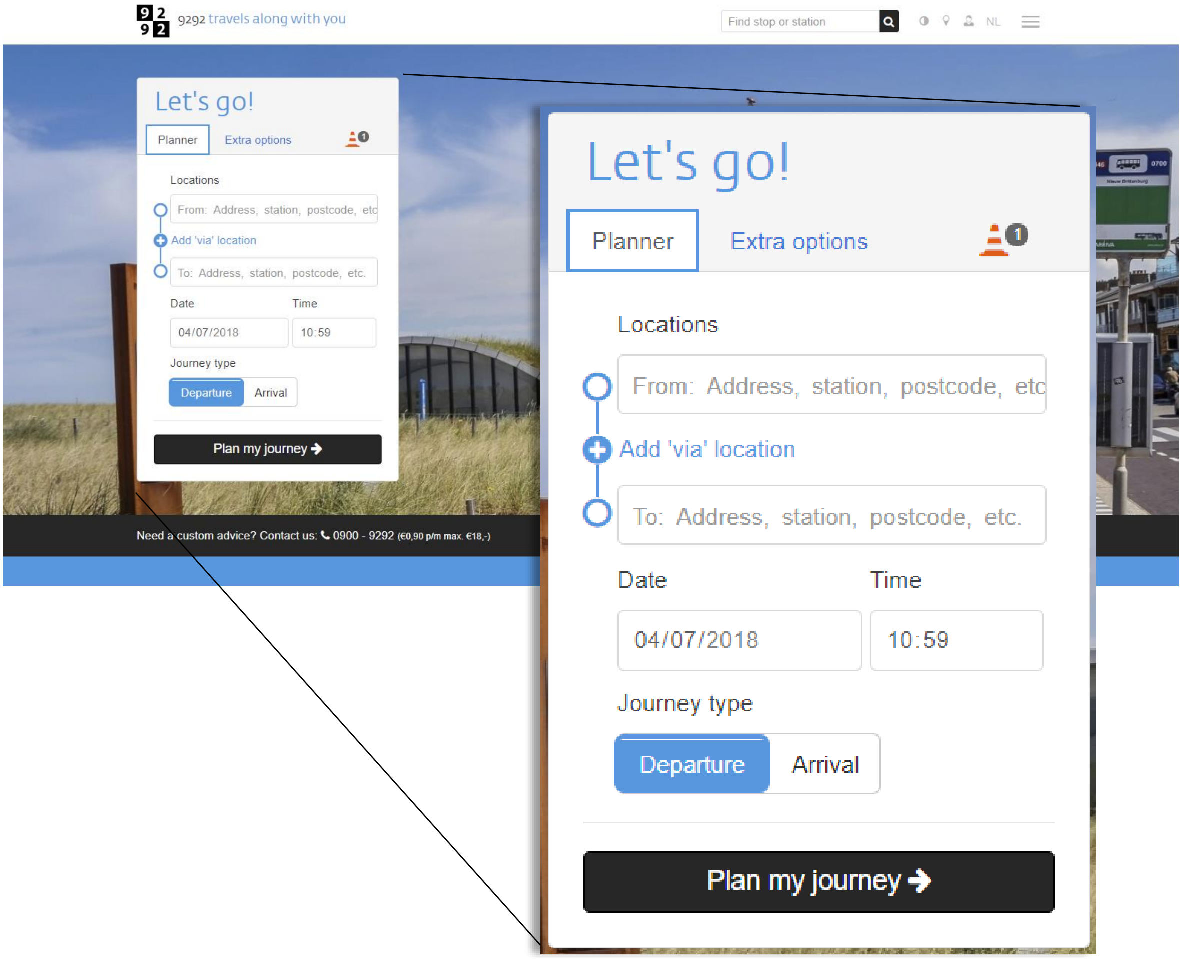
The 9292-journey planner is an interactive trip planner which is accessible via internet on a web browser (www.9292.nl) or via an app for smartphone or tablet. In the trip planner you can plan a trip by public transport for all modes in the whole country. The planner requires an arrival or departure time and a start and end location as input from the user, see Figure 4.6 for the web-based version. The planner then searches for the most suitable (multimodal) journey by combining and comparing the public transport supply. The most suitable journey and its alternatives are presented to the user via the interface as shown in Figure 4.7.

The best fitting journey is the first reasonably fast journey that starts after the time set as departure time or ends before the set arrival time. This journey is shown in an interface which also lists a few alternatives. These alternatives are presented as summaries with the departure and arrival time, the number of transfers and the total travel time. It is also shown if there are any delays or disruptions.

There are some extra options. The most apparent extra option is to add an intermediate destination. The trip planner will then plan a journey between the start and end destination via this intermediate destination. It is also possible to request 5 minutes extra transfer time, to exclude a travel mode (bus, train, light rail, metro or ferry) or to request for a journey that is wheelchair friendly.

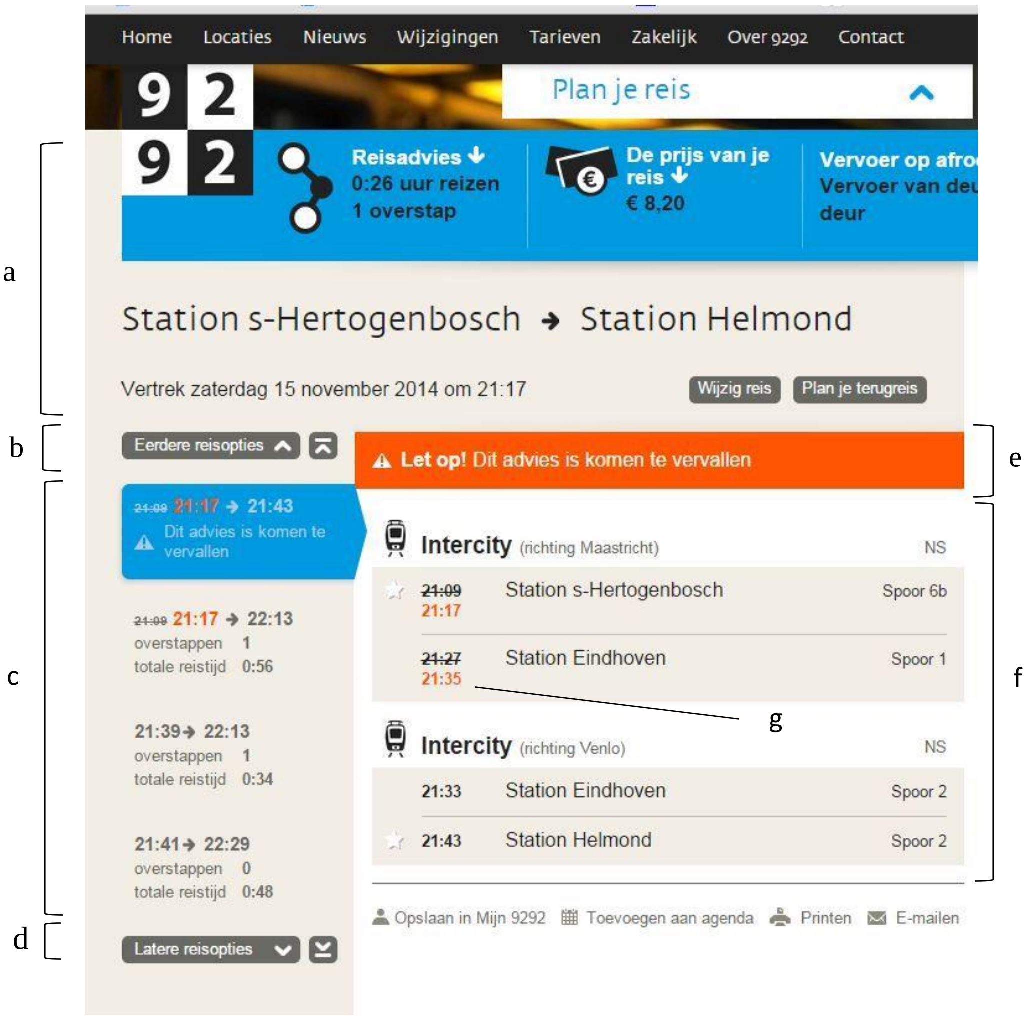
Mulley et al. (2017) found the type and the use of travel information differed with the type of passengers, age and stage of the journey. E.g. older people are less aware of the available travel information sources. They also show that frequent travelers are more aware of the available information sources.

It should be noted that each user uses 9292 in their own manner. It could be that instead of using the via option, users plan their journey in two parts: First from their start point to the via point and the second journey from there to their destination.



Need a custom advice? Contact us: 0900 - 9292 (€0,90 p/m max. €18,-)

Figure 4.6 Homescreen of the webbased trip planner of 9292 with the trip planner magnified. Retrieved from 9292.nl.



Home Locaties Nieuws Wijzigingen Tarieven Zakelijk Over 9292 Contact

9292

Plan je reis

Reisadvies
0:26 uur reizen
1 overstap

De prijs van je reis
€ 8,20

Vervoer op afro
Vervoer van deu
deur

Station s-Hertogenbosch → Station Helmond

Vertrek zaterdag 15 november 2014 om 21:17

Wijzig reis

Plan je terugreis

Eerdere reisopties

Let op! Dit advies is komen te vervallen

~~21:09~~ 21:17 → 21:43
Dit advies is komen te vervallen

~~21:09~~ 21:17 → 22:13
overstappen 1
totale reistijd 0:56

21:39→ 22:13
overstappen 1
totale reistijd 0:34

21:41→ 22:29
overstappen 0
totale reistijd 0:48

Latere reisopties

Intercity (richting Maastricht) NS

~~21:09~~ 21:17	Station s-Hertogenbosch	Spoor 6b
~~21:27~~ 21:35	Station Eindhoven	Spoor 1

Intercity (richting Venlo) NS

21:33	Station Eindhoven	Spoor 2
21:43	Station Helmond	Spoor 2

Opslaan in Mijn 9292 Toevoegen aan agenda Printen E-mailen

Figure 4.7 Travel suggestions in the web-based version of the 9292-trip planner. The interface has the following information blocks: a – the general travel information of the current option; b – button to show one option earlier; c – the four best results given the requests or after the user requested earlier (b) or later (d) options; d – same as b but for later options; e – extra information ("This travel option is no longer viable"); f – the detailed travel plan of the current active option; g – the current and planned times are shown both. Retrieved from http://www.treinreiziger.nl/wp-content/uploads/2016/10/9292-reisadvies4a.jpg.

The motivation of the traveler for using the trip planner causes some typical (noise) patterns in the data. For instance, a potential traveler can use 9292 to see if his travel plans are even possible using public transport, or at what time the first and latest options are. In these cases, the potential traveler is only interested in the availability of a public transport connection at the time he or she needs it. This user will then enter a late (or early time) or use the current time and scroll through all the alternatives to find the latest (or earliest) departure time, whatever method he or she finds most convenient. At a later stadium this traveler might return to the planner to plan his journey in more detail. The requests in this case may be distinctive in time, where the check for possibility happens more than a day in advance and the detailed request happens a few hours in advance. However, in some cases this might be not the case.

Other purposes of using the trip planner are:

- To find the best possible travel plan (which minimizes the weighted utility cost function).
- To recheck an already chosen travel plan. For instance, to check their transfer times, transfer platforms, lines to transfer to, etc.
- To check if their delay causes problems later on in their multimodal journey.
- To check before leaving if there is any delay or disruptions.
- To find at what time the bus leaves every hour. Or to see if there is even an hourly pattern.
- To look if public transport is a competitor for transport by car.
- To adjust their travel plan during the trip because of disruptions or delays.
- To look up historic journeys for declaration purposes.

Thus, based on the intended use, the number of requested alternatives and the interval between request time and departure time differs (well in advance, just before the trip, on trip, afterwards). Because the objective is to predict the number of people boarding in advance, we only can use the requests made pre-trip. This check has to be done per part of the multimodal journey. For instance, if a travel advice consists of two journey parts connected by a transfer, the request can be made on trip for the first part and pre-trip for the other.

It should be noted, that you can classify the travel requests in different ways. One way is by the objective of the user, another could be by looking at what stage in the trip the traveler boards the bus. If the traveler first has to travel 200 kilometers by 3 different trains, there is a chance that he misses a transfer or that the train is delayed. In this case the ridership of this particular bus is dependent on the trains by which it is connected. So there could be less predictive power in requests in which the bus is the last leg in a multimodal/multivehicle trip. Furthermore, the predictive power in a trip with the mode bus as the first leg could be as high as a trip in which the bus is the only motorized mode.

Thus, different kinds of noise are bound to occur in the dataset due to the design, usage and human error.

4.3.2 Data collection

Each time a travel plan is shown (e.g. after the initial search or each time the user selects an alternative), data are logged. For each such occurrence, the characteristics of the total journey are stored as are the characteristics of the sub parts. Figure 4.8 shows an example of the difference between the total journey and the journey parts. These data are logged in two separate tables – *journeyquestions* (Table 4.3) and *journeyparts* (Table 4.4) - which are linked with the field *question_tulp_id*. Thus, data is available for the chained trip as well as for the individual parts.

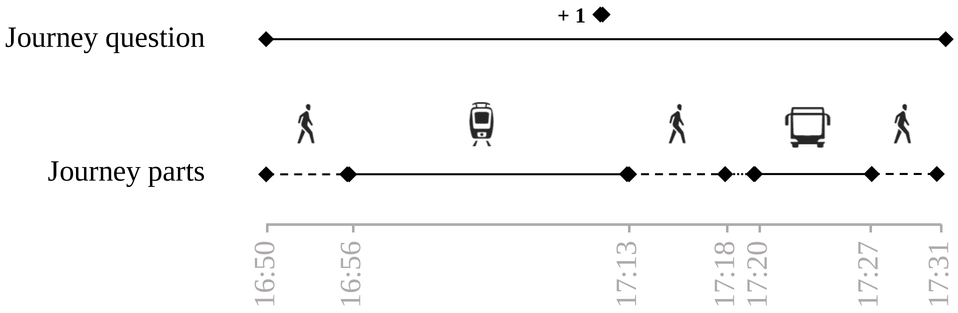


Figure 4.8: Difference between a journey question and its journey parts. For the example the actual advice for a journey between the office of OV-bureau in Assen and the office of Qbuzz in Groningen is used.

The data acquired to perform this research is collected from the APP, the website and the API. The custom requests made by travelers by calling the 9292-call center are also present because the operatives use the website to gather the travel recommendation. The API is used by some other online travel planners like the travel planner from Qbuzz. These other travel planners can have a different design or a somewhat different functionality. This can result in other typical usages and thus data characteristics.

Some of these data are made available for this case study. The data are supplied by 9292 via two connected tables which are displayed in Table 4.3 and Table 4.4. Appendix C shows a raw sample from these two tables.

Field	Type	Description
question_tulp_id	Text	A GUID (Globally Unique Identifier) to denote the question.
planner	Text	
action	Text	
request_datetime	Date with time	The date and time the request was made rounded to minutes.
departuredatetime	Date with time	The date and time of departure for the suggested (multimodal) journey.

Field	Type	Description
arrivaldatetime	Date with time	The date and time of arrival for the suggested (multimodal) journey.
question_type	Number	Indicator if the request was made for a journey arriving before or a journey departing after a certain time: *D* for departure and *A* for arrival
from_halteclusternumber	text	Id of the origin stop or station.
to_halteclusternumber	text	Id of the destination stop or station.
via_halteclusternumber	text	Id of the via stop or station, blank if there was no via option set.
from_halteclusternumberlist	text	A set containing the possible origin stops and stations.
to_halteclusternumberlist	text	A set containing the possible destination stops and stations.
via_halteclusternumberlist	text	A set containing the possible via stops and stations.
no_of_changes	Number	The number of transfers in this journey.

Table 4.3: The variables of the table containing the suggested 9292 journeys

Field	Type	Description
question_tulp_id	text	A foreign key to the overall journey this question was a part of.
journeypart_sequence_no	number	A number representing the order of the journey parts within the journey. Together with the question_tulp_id the
transport_company	text	The transport operator.
line_no	text	The bus line number as known to the public.
transport_type	text	The mode of transport.
start_cluster_number	text	Id of the origin stop or station.
end_cluster_number	text	Id of the destination stop or station.
travel_time	text	The travel time in minutes.

Table 4.4: The variables containing the journey parts of which the journey consists

4.3.3 Data preprocessing

Like the bus data dataset, the trip planner dataset is checked for errors and noise, trimmed and augmented.

Noise that can be expected from the fact that it is not clear from the dataset if the information is requested for an individual or if that individual is making travel plans for a larger group of people. In which case it is harder to predict the number of passengers boarding a bus. Furthermore, it is inevitable that some logged travel requests are the result of users making a mistake when entering their travel request or just mis clicks. Of course, the mis clicks are also dependent on what type of device is used and the condition of the device. For instance, on average the screen of a smartphone is smaller than the screen of a tablet and pc, therefore it is likely that unintentional clicks are more prone to happen with smartphones, even more so when the touchscreen is cracked. In such cases, the request is logged but might not intended to be consulted by the traveler. It could be that an intended consult and an unintended consult can be separated by the time the advice is viewed and the number of requests the user made that session. However, at the moment these two noise sources are not detectable.

We are mostly interested in the journey parts since they comprise the potential trips made per bus. However, the data on the total journey and the non-bus journey parts can be used to augment the records of the bus trips. This could be valuable information since an individual part can be favorable whereas the characteristics of the total journey are not.

The start time and end time per journey part are unknown, they are only logged for the total journey. We augmented the journey part table by adding the reconstructed start and arrival time. These times are reconstructed by summing the travel times of prior journey parts to the journey departure time. Unfortunately, this method introduces some noise, since the travel times do not incorporate waiting times. We validated the results by checking if the last arrival time of the last journey part matches the arrival time of the overall journey. Upon validation, 43,646 journeys (0.47%) stood out because the travel times did not add up to the journey travel time. These time differences range from 1 to 401 minutes.

As stated before, we have to judge if a request can be used for the forecast (if the request is made pre-trip) per journey part. Therefore, this table is augmented with a field containing the request interval in minutes. This feature tells how many minutes there are between the travel information consulting and the start of the journey part. This feature is constructed using the *request date time* from the journey table and the newly constructed field *start time* from the journey parts table.

Some records had to be discarded; The journey parts table initially contains 67,009,344 records. We will discard the records with a depart time before 31-12-2016 23:00 and arrival time after 01:00 on 15-04-2017 and the data of journeys with the only Qbuzz parts not starting or finishing in the study area. Furthermore, we discard parts belonging to a journey that took more than 24 hours or that belong to a journey that was requested for the past. Journey parts that do not correspond with a bus trip by Qbuzz are not discarded but neglected. After trimming there are 11,694,849 records left.

4.3.4 Data exploration

There are 3,092,124 travel requests (of the 15,586,643 unique ones) that have an interval of 0 minutes between the request time and the departure time (both times are rounded to minutes by 9292), see Figure 4.9. It is not very likely that 30,9 % of the travel requests are consulted at exactly the same time the traveler has to depart. This could be an error in the dataset. Since we want to predict the ridership more than 0 minutes in advance, we have to disregard most of these data. As stated before, the interval time is calculated for journeys and not for the trips. Therefore, it could be that interval time for the Qbuzz leg is bigger as the prediction interval, while the interval time for the journey as a whole is not.

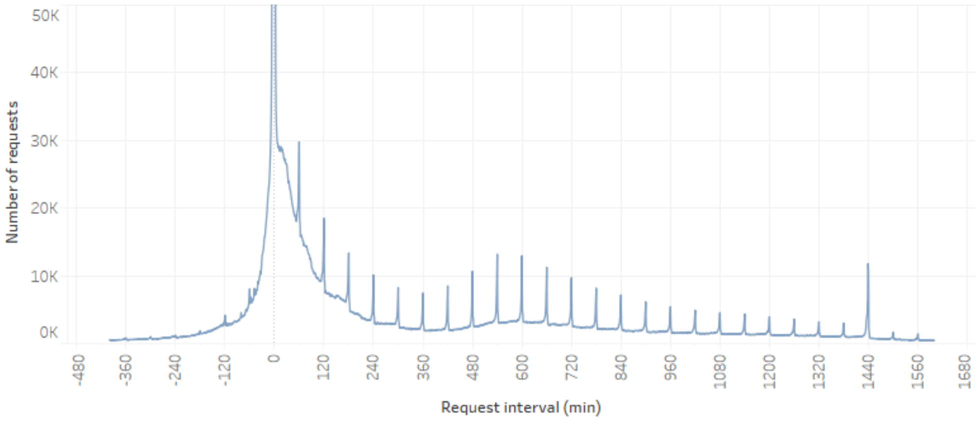


Figure 4.9 Requests per request interval ; e.g. the time between making the consult and the suggested departure time

If you look at the request interval, as depicted in Figure 4.9, you can find the largest peak around zero minutes (39,9 % of the requests had a request interval between -2 and 2 minutes). The same figure shows a pattern on the positive x-axis of peaks around the multiple of 60. This could be caused by the design of the journey planner. The user gets the chance to set the time. And when doing so, it is easier to only set only the hour around the time you want to depart or arrive and leave the minutes at the original value. You can see a similar phenomenon at the peak of 1440 minutes (1440 minutes is 24 hours), which is caused by consulting a travel plan for exactly one day after the request time, in that case the traveler only adjusted the date.

Around the 600 minutes there is a slight increase in the average number of requests. This is largely caused by people who consult the day before in the evening a journey for the next morning.

There is also a pattern in the departure and arrival time of the consulted travel requests. Figure 4.10 shows the number of requests aggregated per minute of the hour (neglecting the hour and date) for the whole period. If the travel request was set at a given departure time the departure time of the suggested journey is taken into account and vice versa. From the graph we can conclude that, although the total volume is less, the arrival requests has a similar trend as the departure requests. Only the smaller peaks at 18 minutes past and 48 minutes past are missing in the arrival line. In the graph you see a big peak at 0 minutes, a smaller one at 30 minutes and peaks at every 5 minutes interval. This could be because of the system we use with 24 hours in a day and 60 minutes in every hour: It causes less cognitive strain to think of a "round" time: in order of cognitive ease at the hour, half past an hour and at intervals of 5 minutes. The smaller peaks in the departure-time-line at 18 and 48 minutes are likely caused by extra requests by travelers who search a journey which is a connecting service from the intercity (national train) to Groningen which arrives at 13 and 43 past the hour at the station.

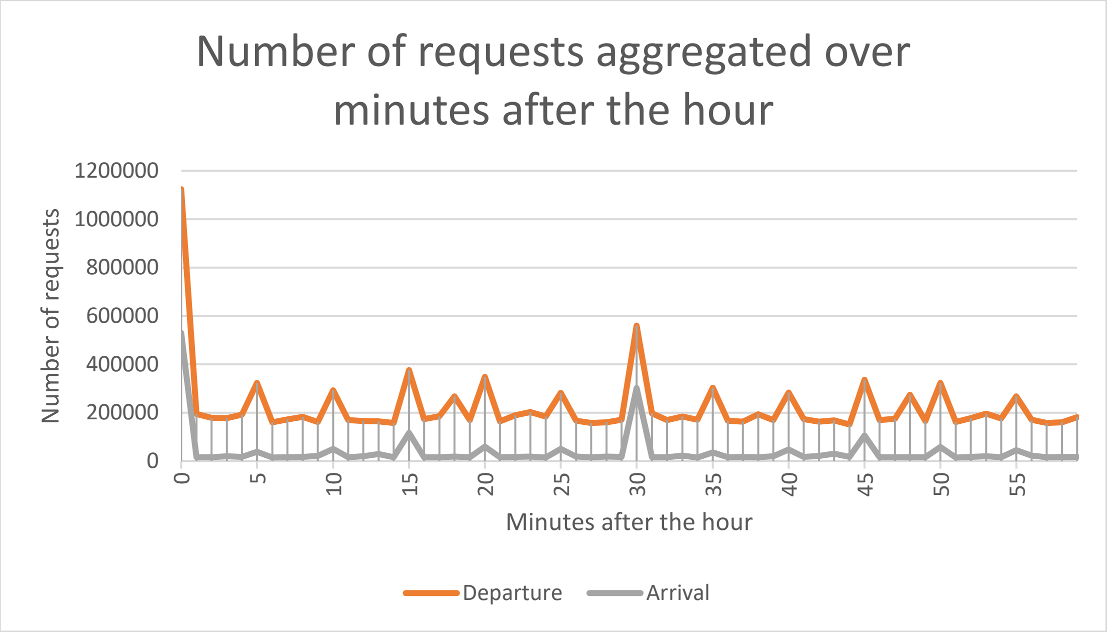


Figure 4.10 Number of requests aggregated over minutes after the hour

The peaks in the departure time at an interval of 5 are not due to an increase in departing buses at that time. In Figure 4.11 you see no big increase in departures around the interval of 5 minutes. Even if you correct for the number of times the stop is requested, there is no clear pattern. The stops around Groningen Central station were in the cluster list that was most requested for (requested around 16% of the time and 5.5% as often as the 2nd most requested cluster list). Here there are also no clear peaks at the 5 minute intervals.

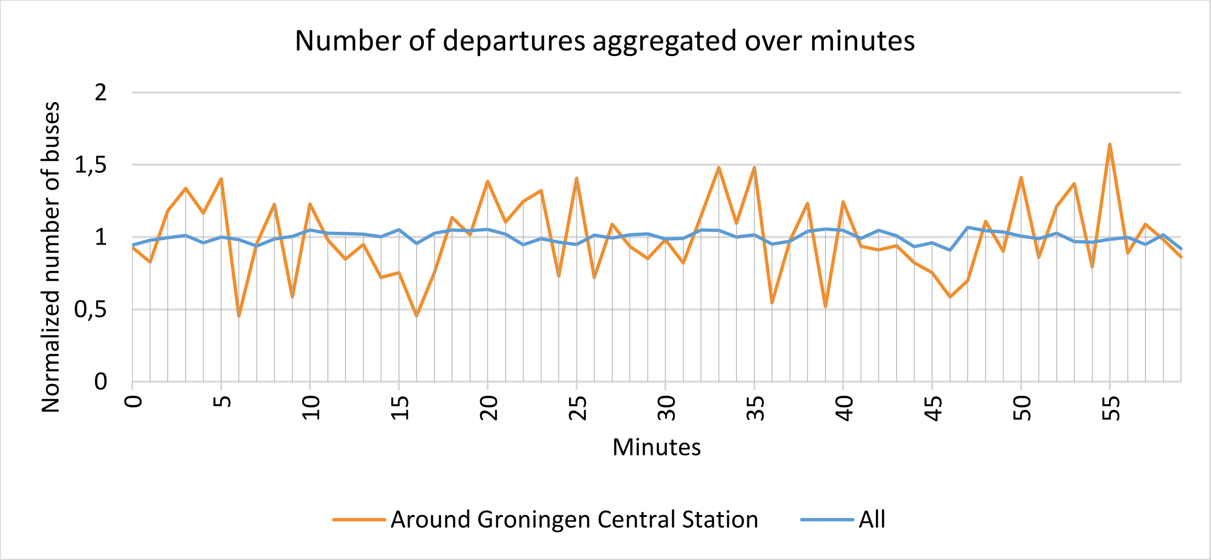


Figure 4.11 Number of departures aggregated over minutes after the hour

4.4 Dataset 3 – Smart card

We will use the smart card data to express the ridership (the realized demand). The ridership is needed to train the forecasting model and to measure the accuracy of the forecast. These data might also be used to derive explanatory variables. For instance, the average ridership of a line based on historic trips can be used as predictor variable.

Smart card data has been used in research before to examine passenger traveler behavior and for ridership forecasting purposes (Ma et al., 2013). Using smart card data has the benefit that it is continuous and it is easy to construct these OD-flows. Furthermore, smart card data shows the revealed preference instead of the stated preference of travel surveys (van Oort et al., 2015b).

Smart card data are chosen over surveys because it is less time consuming to collect and less budget consuming than surveys (van Oort et al., 2015b), it provides better insight into revealed passenger behavior (van Oort et al., 2015b) and it is easier to get a larger sample space and identify frequent travelers (Bagchi & White, 2005). Moreover, you can aggregate smart card data to any necessary spatial or temporal level (Li et al., 2014). However, important factors like the purpose of the journey, are not included in these data (Bagchi and White., 2005). Also, the smart card data do not account for other payment methods and fare dodging.

4.4.1 Description

Since 2012 the OV-chipkaart (the Dutch smart card) is adopted as main fare system for all public transport modes in the Netherlands (van Oort et al., 2015b). Travelers have to first acquire a personal or anonymous smart card and deposit some money on this card before they can travel using public transport.

The usage of the card changes depending on the public transport mode. For bus and light rail, you check-in when you board the vehicle and you check-out before alighting. Because data is available for each time the traveler transfers, it is easy to construct the exact journey the traveler has made using bus and light rail. This task is more complex for other modes like the train and metro, since you only have data of the origin and destination of the total journey.

However, this dataset has some drawbacks: The current privacy law dictates that these individual logs may only be kept for a limited amount of time. Otherwise you could construct a personal profile from this big data (Van Oort et al., 2015b). In addition, because of the current tender system in the Netherlands where operators are competitors and margins are small, the transaction data are regarded as confidential company information. That is why these data is limited available (Van Oort et al., 2015b).

The smart card dataset is not all encompassing. It is still possible to buy paper tickets for the bus, although these tickets are more expensive. Trips made with these paper tickets are missing in this dataset. Also, trips made by people who (un)intentionally dodge the fare are not recorded in this dataset. Another possibility is to partially dodge this fare (un)intentionally by only checking in or out. Dodging the fare (partially) is not allowed, however people do it for the financial benefits, due to forgetfulness or because of malfunctioning equipment.

OV-bureau estimates that 5% of the journeys are made with a paper ticket and 2% of the fares are dodged. The exact distribution of these trips over time and space is unknown. The partial dodged trips are recorded in the dataset. Before preprocessing around 1.56 % of the transactions had no check out location registered. For these trips only one of the locations is known. This location could be the origin or the destination, since something could have gone wrong while tapping in or out. These records are therefore considered noise and removed from the dataset.

The smart card data does not contain the purpose of the trip. Other researchers encountered the same problem. Choi et al. (2012) suggests that the best way to guess the trip purpose is by looking at the time of day. Other methods to categorize trips are by ticket type (van Oort et al., 2015b), by the frequency of travelling on the same corridor on certain times or by the overall frequency.

4.4.2 Data collection

In the Netherlands, Translink is responsible for processing the transaction data. These data are collected locally and send to Translink as schematically shown in Figure 4.12. These transaction data are stored in a database.

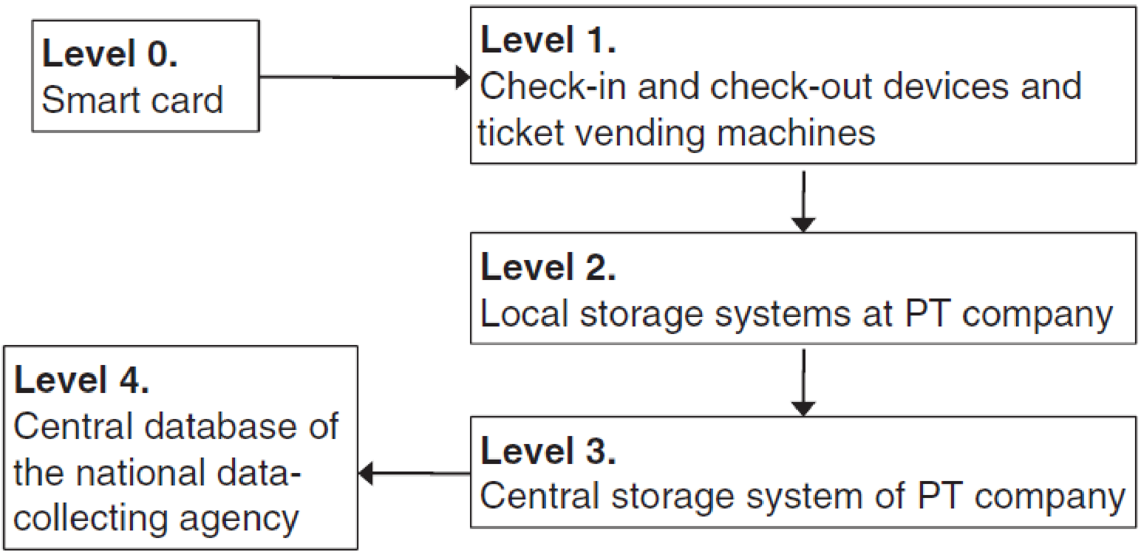

Level 0.
Smart card
Level 1.
Check-in and check-out devices and ticket vending machines
Level 2.
Local storage systems at PT company
Level 3.
Central storage system of PT company
Level 4.
Central database of the national data-collecting agency

Figure 4.12: Flow of the smart card transaction data. Retrieved from van Oort et al. (2015b).

In this database there is a record for each smart card each time a transaction takes place. There are multiple types of transactions. The three most frequent ones are checking in, checking out and depositing money on the smart card.

This dataset has a flaw: the bus trip or bus line is not known. There is an identification number for the used card reader (tap in/out device) but there is no dataset available which matches the card readers with the vehicle registration numbers. Therefore, the smart card data has to be matched with the bus data. Because major bus stations have multiple buses boarding/alighting/departing at the same time, it is not possible to use these data directly: the individual check-ins and check-outs cannot be matched with an individual bus trip. Aggregating the check-ins and check-outs into trips makes matching with unique bus trips possible: It is not likely that two buses have the same departure and arrival times at the same stops.

Thus, the transactions are aggregated to a trip level. Check-in/check-out pairs are constructed by finding the next check out per check in within an interval of 150 minutes. The 150 minutes limit is valid, since no bus trip has a duration larger than 150 minutes. Another option would be to find the closest earlier check in per check out (check-ins without a check-out are neglected) and construct a trip.

We will use unlinked trips in this study. Chaining trips into journeys like Ma et al. (2013) would have the benefit that you could compare the smart card dataset and the 9292 dataset at a higher level as well. However, constructing these journeys is a complex task in itself. Furthermore, doing this 100% reliably would have been nearly impossible. To construct journeys, we would have to use a rule to identify transfers. Most likely this rule would involve some time constraint between two consecutive smart card trips. Using such a rule would introduce some errors. The maximum transfer time should be large enough to make transferring between two non-frequent lines possible. On the other hand, it should not be too large, otherwise you would falsely identify a transfer where the traveler needed less time at his destination before going to his next destination. Furthermore, the alighting stop can also be the boarding stop during a transfer. But as Van Oort et al. (2015b) mentions, it could be that the two legs are connected by a short walk. Also, we only have smart card transaction data of Qbuzz, therefore if a traveler transfers to another operator, it is not at all possible to construct the real journey of the traveler. Because of these reasons we decided not to pursue the comparison on a higher level.

Since we only use the transaction data to express the ridership, we only need a few fields. However, as stated, it might be useful to gather some extra information about the bus lines, for instance the average number of frequent travelers using a bus service or the average type of traveler (student, e.g. with or without subscription). Therefore, we will use the following 7 fields:

- END_TXN_TIME – The real transaction time
- CARD_ID – A unique identifier for the smart card
- CARD_TXN_SEQ_NUMBER – A sequential number representing the order of the transactions
- TXN_SUBTYPE - The transaction type
- START_LOCATION – The start location of the transaction
- END_LOCATION – The end location of the transaction
- Subscription type – A number representing the subscription type that is active on the smart card

The initial pre-processing will be done by Translink, since the handled data is privacy sensitive. Since Qbuzz operates in multiple concessions across the Netherlands, an extra requirement was that the transaction started and ended at a stop in Groningen or Drenthe. Thus, trips traversing the study area boundary are neglected altogether regardless the length of time they travel in the study area.

Per trip different variables are stored as shown in Table 4.5 (Appendix C shows an unprocessed sample). First an id, as well as the starting point, the destination, the departure and arrival time are retrieved (respectively *the id*, *cki_location*, *cko_location*, *cki_datetime* and *cko_datetime*). Additional information like the time block and the active subscriptions on the smart card used for that trip are recorded (*tijdsblok* and *product*). The features *totaal* and *frequentie* record how many times the smart card has made a trip and how many times the smart card has taken the exact same trip (departure times match or are in the same time block, as well as a match between the origins and the destinations).

It should be noted that there are no fields containing information about the line or trip number. Translink does not have these data. We will later try to retrieve the bus line and trip number ourselves. This process is described in section 4.6.3.

It could be that the data are corrupted. For instance, the check-in and check-out times are determined locally and thus are wrong if the time settings of the bus were off. Large errors, bigger than 5 minutes or half the headway, will cause bad merging behavior later on, see section 4.6.3.

Field	Data	Construction method
Id	a number as unique identifier of the trip	This serial number is generated.
cki_datetime	The date time at the check-in	The *END_TXN_TIME* recorded at the associated check-in record.
cki_location	a number representing the check-in stop	The END_LOCATION of a record with a *TXN_SUBTYPE* set to check-in.
cko_datetime	The date time at the check-out	The *END_TXN_TIME* recorded at the associated check-out record. Null is assigned, in case of no check-out.
cko_location	a number representing the check-out stop	The *END_LOCATION* of the check-out (*TXN_SUBTYPE*) which comes after the associated check-in following the *CARD_TXN_SEQ_NUMBER*. (*END_LOCATION* of the check-in and the *START_LOCATION* of the check-out should match.) Null is assigned, in case of no check-out, e.g. if a second check-in proceeds the check-out.
Product	a number representing the subscription on the card	Extracted directly from *Subscription type*.
tijdsblok	A string representing if the trip was made during the weekend, peak hours or of peak.	Constructed from the *cki_datetime*. Dates in the weekends are assigned the value weekend. The other trips are during morning peak if the time is between 05:00 and 10:00, evening peak for trips between 14:00 and 20:00 and off peak for all others.

Field	**Data**	**Construction method**
frequentie	the number of same trips registered in this dataset with this smart card	This number is calculated as follows: Construct for each *CARD_ID* the distinct journeys, e.g. journeys that do not have the same starting points and destinations or that have departure times that are not in the same *tijdsblok*. Count for each of these distinct journeys the number of occurrences this is the *frequentie*.
totaal	The total number of trips registered in this dataset with this smart card	Count for each *CARD_ID* how many trips are made.
is_student	Boolean representing if the smart card has a student subscription on it	Extracted directly from *product*.

Table 4.5 The collected fields from the smart card data

4.4.3 Data preprocessing

In total there are 14,432,812 records between 04-10-2016 and 16-06-2017 in the smart card dataset. However, before 01-11-2016 the data is incomplete. During the trip constructing by Translink, some exception codes did arise and were recorded. In total there are 5 codes present in the dataset, see Table 4.6.

Exception	Description	Count
0	No exception	14154988
4	There is no check out	225433
5	The check out is at the same stop id and within 10 minutes of the check in	33171
32	The check out is at the same stop id and between 10 minutes and an hour after the check in	18964

Exception	Description	Count
33	The check out is at the same stop id and after more than an hour after the check in	256

Table 4.6: The exception codes, the occurrence frequency and the description

The 1.6 % stops without a check out location could be travelers who (un)intentionally partially dodge the fare or because the traveler used this stop to travel to a destination outside the scope boundary of Groningen and Drenthe. For this analysis we will discard these data.

The same check out stop as check in stop (exception codes 5, 32 and 33) could denote a passenger boarding a bus and changing his or her mind. It could also denote a passenger making a round trip (the 9292 dataset has also round trips as travel suggestions). Nevertheless, we will remove this data from the dataset.

We will also discard the data for which the check in location and check out location is not present in the other datasets and we will discard the data before 31-12-2016 23:00 and after 01:00 on 15-04-2017 and transactions with an earlier check out time as check in time. Finally, the records are augmented with the corresponding 9292 cluster resulting in a trimmed dataset of 6,814,907 records.

4.4.4 Data exploration

Figure 4.25 shows the average number of smart card trips starting per hour of day. This figure shows a clear distinction in pattern between the 4 schedule date types. The weekdays outside the holidays have two peaks, where the morning peak (around 8 AM) is sharper and more clearly defined as the evening peak (around 4/5PM). During the weekends there are no peaks and overall less trips. The shape of the distribution is more like a dome with the highest clearance in the midday and evening. The weekdays during small holidays show a distribution that looks like mixture of the two. There is a morning and evening peak, but they are less defined and the evening peak is larger than the morning peak. Furthermore, the distribution is overall shifted more to the midday and evening.

Figure 4.13 shows the number of check ins per day. The weekends (green circle) are clearly visible from this figure, since the total number of check ins is much lower (around 25 – 15k). Also, the two holiday weeks (orange circles) are visible because the drop in demand. Furthermore, some weeks show some intraday patterns between the weekdays.

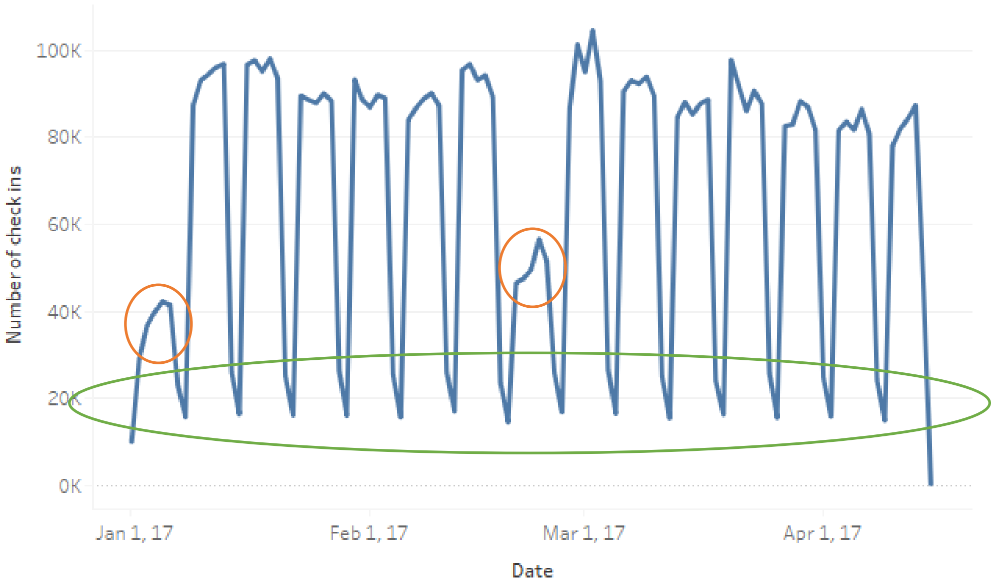


Figure 4.13: The number of check ins per date

The smart card dataset lets us also investigate the number of frequent travelers. Figure 4.14 and Figure 4.15 show the cumulative distribution of the trip frequency of a smart card. These figures show an exponential distribution with many people with a relative low frequency and a few with large frequencies. It should be noted that these figures are made with the features *frequentie* and *total* which are measured for the untrimmed dataset. The untrimmed dataset spans a longer period of time and contains 32.6 weeks (instead of 13 weeks).

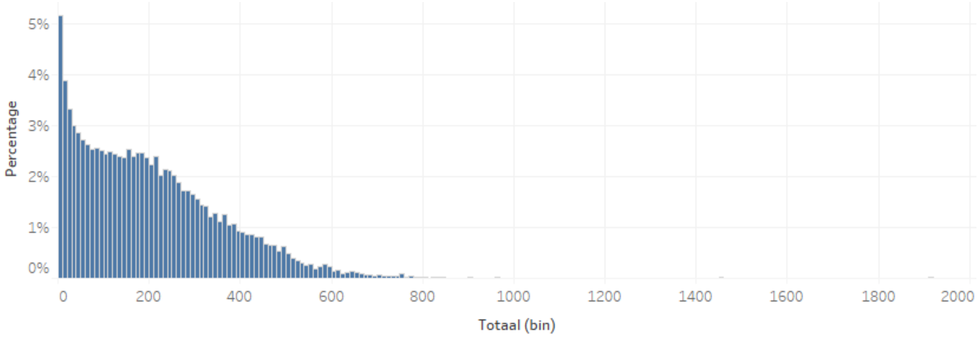


Figure 4.14: The distribution of the frequency the smart card is used for a(n unchained) trip with Qbuzz

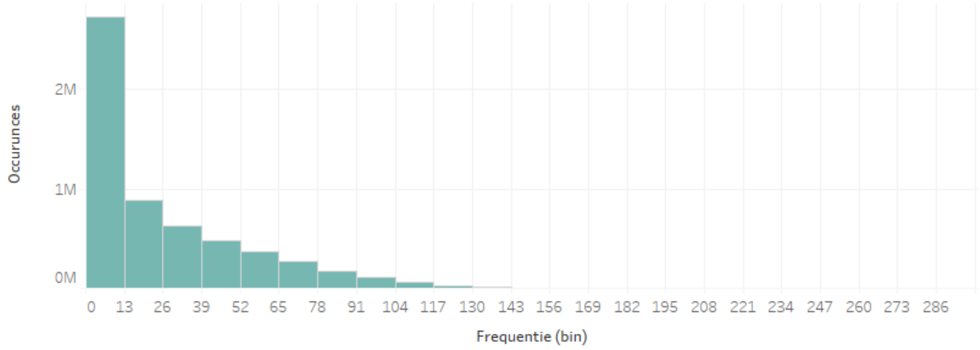


Figure 4.15: The distribution of the frequency the smart card is used for a similar trip ; e.g. same time period and same corridor

4.5 Dataset 4 - Rain data

From the literature review it was clear that weather had an influence on ridership. Some studies even found that travelers sometimes choose a different departure time last minute due to rain. Thus, it could be that due to rainfall the explanatory power of the trip planner data changes. Therefore, we will include information on this data.

4.5.1 Description

We will use data from the Royal Netherlands Meteorological Institute (KNMI). The KNMI collects daily and hourly weather data from different stations around the Netherlands.

4.5.2 Data collection

We downloaded a dataset containing the hourly summed rainfall and rain duration from http://projects.knmi.nl/klimatologie/uurgegevens/selectie.cgi. We chose to download data from 8 weather stations which are inside the study area or closely around it, see Figure 4.16.

The hourly sum of precipitation is stored as an integer which tells how many 0.1 mm rainfall there was during that hour. The duration of the precipitation duration is also recorded as an integer which tells the number of 0.1 hours rainfall for that hour.

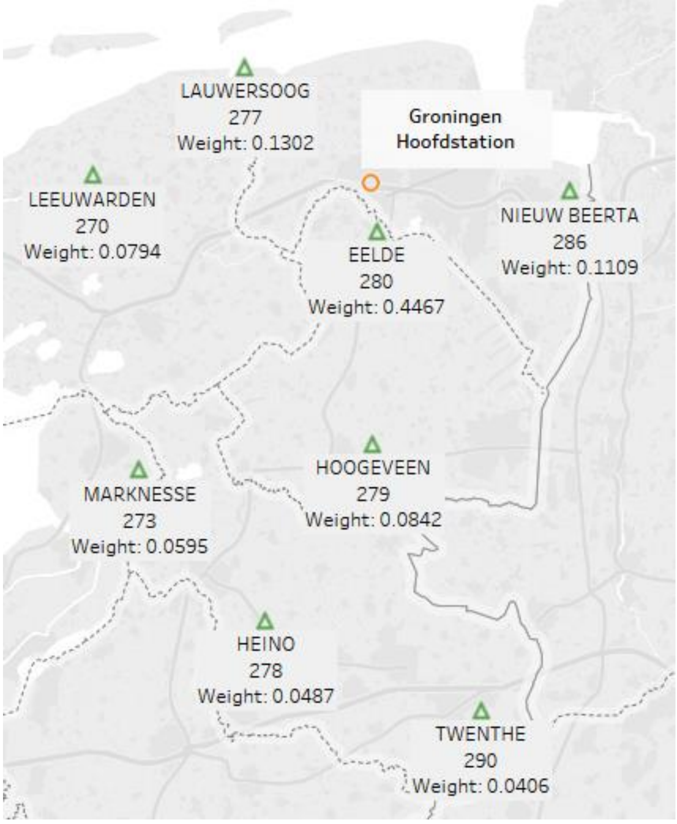


Figure 4.16: The 8 used weather stations and their id with the weights for bus stop Groningen Hoofdstation

4.5.3 Data preprocessing

Only one step is needed to prepare the dataset of the KNMI. For instance, the hourly sum precipitation is originally a combined quantitative and categorical variable: when there was 0.05 mm recorded this was logged as -1 instead as 0.5. In order to correct this, we changed the variable type to double so we could change the records with -1 to 0.5.

4.5.4 Data exploration

Figure 4.17 shows a sample of the rainfall and duration of the rainfall for two weather stations. There are time periods where both weather stations had some rain. However, upon zooming in on the data there are some differences in the timing of the rain and intensity. Furthermore, there are also time periods where there is only a peak in rainfall and duration at one of the weather stations. Figure 4.18 shows the distribution of the rain duration. From this figure we can conclude that during most hours there is no rain. Furthermore, there is a small peak at 60 minutes, so some rain showers last longer than an hour.

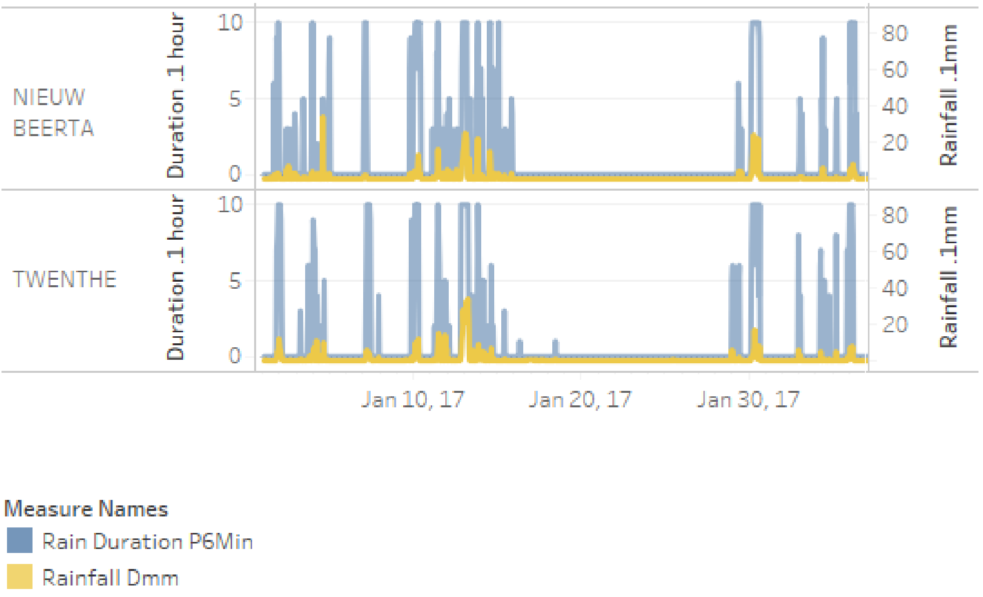


Figure 4.17: A sample of the rainfall and duration of the rainfall per hour for 2 weather stations.

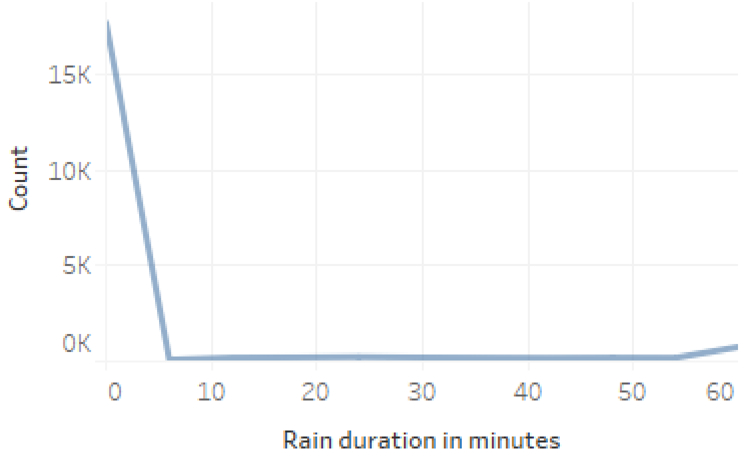


Figure 4.18: The distribution of the rain duration. The rain duration is stored per 6 minutes.

4.6 Data fusion

This section discusses the merging of the 4 datasets. The goal of the merge is to relate the trip planner trips and the smart card trips with actual bus trips (according to the bus data dataset) in order to create a final dataset similar to Table 4.1. Before we match the travel requests, transaction data and bus data, we needed to match the stops and the dates.

We already discussed the synchronization of the datasets on time (adding a timestamp to the journey parts and a date to the 4 time variables in the bus data dataset). However, we have to do some steps in order to match the stops. These steps are discussed in section 4.6.1. Afterwards, we can merge the datasets on stops and time together.

4.6.1 Matching stops

The datasets use a different kind of aggregation level for the bus stops. OV-bureau and Translink make use of the stops as stored in the *Centraal Halte Bestand* (CHB). CHB keeps a record per platform. Thus, if a stop has multiple platforms, these are stored as multiple stops. Most stops have two platforms, one for each direction on opposite road sides. 9292 aggregates these stops to clusters, where one cluster is the grouped collection of the platform belonging to the same stop.

The stops are matched by first aggregating the stops of the CHB to clusters and then match the stops on name and distance. The few clusters that could not be matched this way or had a large distance were later matched by hand. The full report of the stop matching can be found in Appendix F.

4.6.2 Matching trip planner data with bus data

This section describes the trip matching of the trip planner trips with actual bus trips. This is a 6-dimensional problem, where we try to find the best match within these dimensions, see Figure 4.19. For example, when we match a journey part to an actual bus trip:

1. The suggested boarding stop (origin) should be serviced by the bus trip
2. The suggested alighting stop (destination) should be serviced by the bus trip sometime after the boarding stop was serviced
3. The suggested boarding time should be at the same time the bus dwelled at this stop
4. The suggested alighting time should be at the same time the bus dwelled at this stop
5. Depending on the request interval time, the recorded or the planned time should be used to determine the possible match
6. The suggested line number should be the same as the public line number of the actual bus trip

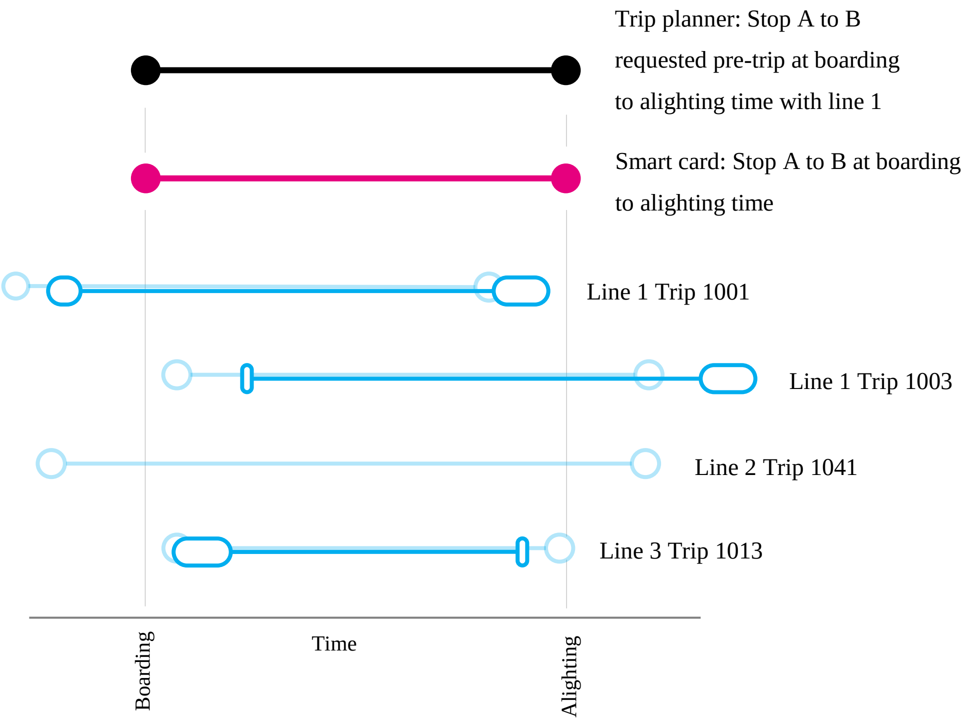

Trip planner: Stop A to B requested pre-trip at boarding to alighting time with line 1
Smart card: Stop A to B at boarding to alighting time
Line 1 Trip 1001
Line 1 Trip 1003
Line 2 Trip 1041
Line 3 Trip 1013
Boarding
Time
Alighting

Figure 4.19: The 6 dimensional problem of matching the request to an actual bus trip . Black: the trip as requested by trip planner. Blue: trips as in the bus data dataset which could be a match based on the OD pair with in light the planned time and in darker blue the recorded times. The circles/rounded rectangles represent dwell times. Based on the dimensions origin and destination there are many matches for which five good ones are shown. Based on the dimension of public line number the bottom 3 matches do not match. Based on the dwell time constraint there is no match at all. For smart card trips the problem is 4 dimensional since no line number is available. Thus, the matching problem for trip planner is more constrained.

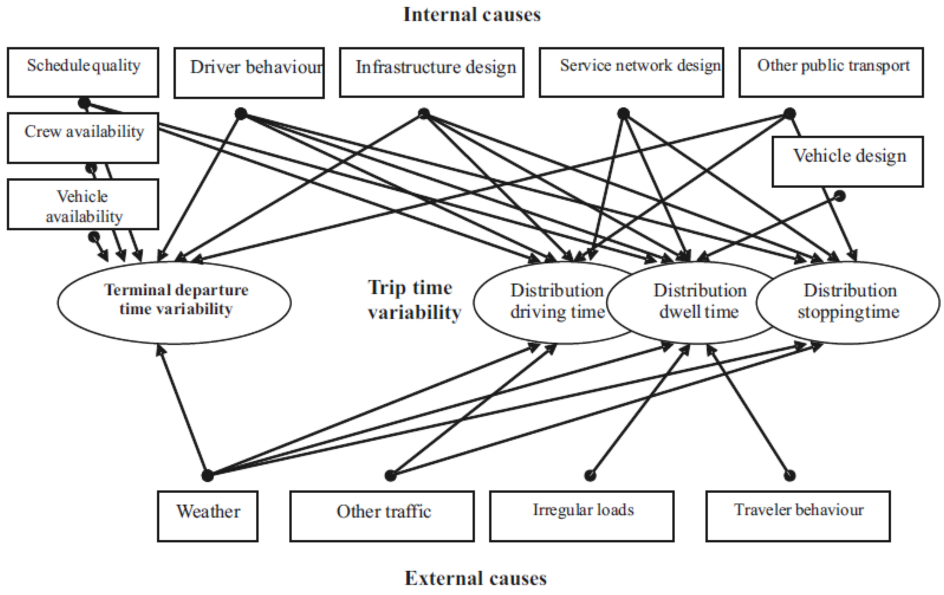


Figure 4.20 Causes for delay . Retrieved from Van Oort et al. (2015a)

There is some extra complexity to this problem, because the 9292 trip planner uses the actual departure and arrival times which vary over time (delays are bound to happen because of several reasons, see Figure 4.20): When no current information on the punctuality of the bus (because the bus has not begun this trip yet or the trip is planned a day or more in advance) is available, the trip planner uses the planned times. However, approaching the trip, there will be an actual delay , which is somehow incorporated by the trip planner. Therefore, an advice for departing at 14:02 from *Groningen, Hoofdstation* to *Groningen, UMCG* could correspond to the same bus as an advice which departs at 14:05 with the same origin and destination. The only difference between these two advices is the information the trip planner had on the moment of the request. The suggested departure and arrival times should be dependent on the time difference between requesting and the starting of the bus trip. In the form of a formula:

$$Suggested\ departure\ time_t = Planned\ departure\ time + Delay_{\,t}$$

and

$$Suggested\ arrival\ time_t = Planned\ arrival\ time + Delay_{\,t}$$

In these formulas *t* denotes the time at which the travel information was constructed. We can group the trip request based on this *t* as follows:

Group 1: The bus has not yet started the trip (t < start time of the at the origin of the bus trip minus a margin)

The bus trip has not started yet, thus the trip has not encountered any delays yet. The delays in this group are equal to zero. For this group we utilize the start time of the trip. We include a margin, because sometimes a delay is reported before the bus trip has even started. This happens when no bus is yet coupled with the trip.

Group 2: The bus has not yet departed at the origin stop of the request, but the bus trip has started (t < suggested departure time and t > start time of the at the origin of the bus trip minus a margin)

In this group the bus could have some current delay. The delay for the departure time and arrival time is at most equal to the most current delay known to 9292 which comes from the *koppelvlak 6* messages which are sent by the bus.

Group 3: The bus has departed the origin stop of the request but has not arrived yet at the destination of the request (t > suggested departure time and t < suggested arrival time)

In this group the bus could have some current delay. The departure for this trip lies in the past, so the delay for the departure is static and equal to what was reported via the *Koppelvlak 6* message. The delay for the arrival time is at most equal to the most current delay known to 9292 which comes from the *koppelvlak 6* messages which are sent by the bus.

Group 4: The bus trip lies in the past (t >= suggested arrival time)

The suggested times in this group are suggested after the bus has serviced the destination stop of the request. Thus, the delay is static and equal to the recorded delay as last reported by a *koppelvlak 6* message.

Unfortunately, we only have an abstract of the messages of *koppelvlak 6*, namely the actual passage times of a bus at a stop. In this dataset the intermediate messages are missing -the bus sends messages at an interval of one minute, the interval is even smaller when an action is triggered for instance when the bus is off route (BISON,2014)-, therefore we don't really know what the reported maximum and minimum delay were during the trip. For illustrative purposes we visualized the actual delay and hypothetical suggested departure and arrival times for a hypothetical trip request in Figure 4.21. This figure shows the relation between time, current punctuality and the suggested departure and arrival times. Furthermore, it shows how you can divide the trip planner requests into the four groups using the request interval (globally pre trip, bus trip started, on trip and afterwards). In this example the punctuality is directly added to the departure and arrival time. However, 9292 first does some computations, like interpolations, to incorporate the possibility to lessen the delay.

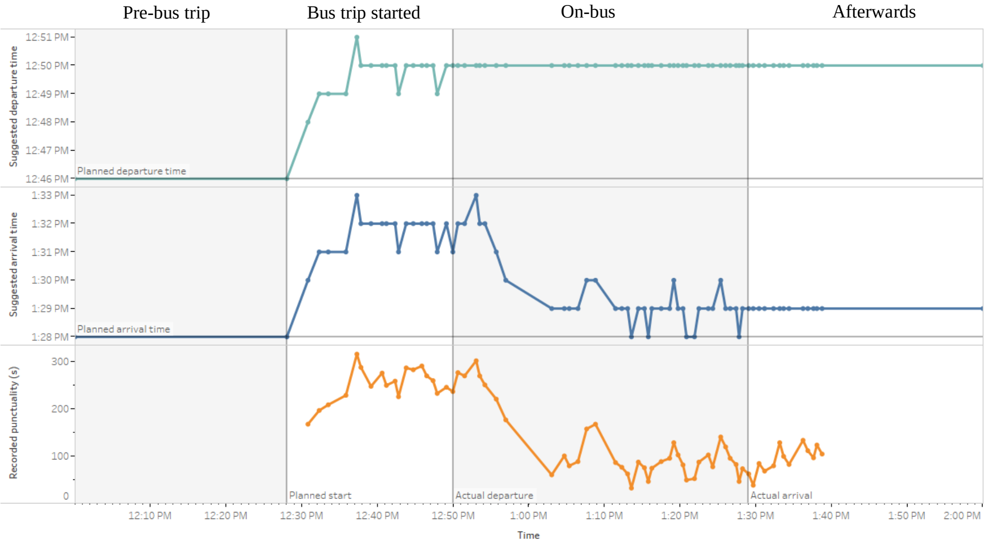


Figure 4.21: Grouping requests based on request interval. The hypothetical suggested departure time, hypothetical suggested arrival time and the recorded delay for trip 1014 of line g039 on 09-01-2017 which in general had a positive delay and a hypothetical information request between Groningen, Leegeweg (19th stop) and Doezum, Kerkplein (57th stop). Each point represents the passing of a stop. The planned start time of the trip, actual departure time of the bus at Leegeweg and the actual arrival time at Kerkplein divide the requests in 4 groups based on the time of request.

Figure 4.21 shows it is harder to match requests from groups 2 and 3, because the used departure and arrival time fluctuates due to delays. Trips of group 1 should be easy to be matched because the suggested departure and arrival time should be the same as the planned departure and arrival time. However, when trying to match the 9292 trips from group 1 (for this case trips which were requested a day or more in advance) with the actual bus trips, 70% of the 9292 trips could not be matched. For example, a 9292 trip starts on Wednesday 01-03-2017 at 14:02 at cluster *1800118 – Emmen, Stadionplein* using line 26. However, there is no trip recorded in the OV-bureau dataset which departs or arrives at that time at that stop for any date.

We will try and match the travel requests to actual bus trips taking these 4 groups into account and using a margin of 15 minutes. Since we are dealing with a real-world dataset there are some complications. For instance, there is no recorded actual arrival and departure time for many stop passages and when there is a time recorded, these times are not always accurate as described in section 4.4.3. Therefore, you sometimes have to base the match on the planned arrival time. Thus, the requests are matched using the following time dimensions:

1. Journey parts from group 1 are matched using the target departure times.
2. Journey parts from group 2 are matched using the average of the target and recorded departure times. If the recorded time is not available, the planned time will be used instead.
3. Journey parts from group 3 are matched using the recorded departure time for the start and the average of the target and recorded departure times for the end. If the recorded time is not available, the planned time will be used instead.
4. All journey parts with the end time before request time (including an additional 15 minutes slack) are matched using the recorded times for both the departure and arrival time. If the recorded time is not available, the planned time will be used instead.

Beside the time we will match the trip planner trips with the bus trips on location and line number, see Figure 4.19. We will perform the trip matching in steps: First we will match the trip planner trips to all possible bus trips during a larger period of time. For this we choose two different time windows. Method 1 has a time window that runs from 4 hours before the planned departure time of the bus at the stop till 4 hours after. The time window of method 2 is smaller and runs from 5 minutes before till 4 hours after. Here we chose only 5 minutes before since it should not happen that the bus passes a stop more than 5 minutes early (so a travel requests that starts 6 minutes before the planned departure time should not be matched). The time margin after the planned time is larger since it is possible that a bus gets delayed. The matches of method 1 will be used to better understand the consequences of using the smaller time window preceding the trip. After rounding up the possible trips, we will pick the best fitting trip regarding a metric. We choose to use the sum of the absolute differences between suggested time and actual departure and arrival time. The used actual time depended on the group the requests belongs to as discussed earlier. The best match minimized the sum of the absolute differences.

Executing the process described above we could match about 75% of the requests, see Figure 4.22. It did not really matter which time window was used for the total number of requests that could be matched. This results in the conclusion that the dataset of OV-bureau is a subset of all the bus trips as executed by 9292 within the borders of Groningen and Drenthe. The 5-minute-before-constraint resulted in a delay of the number of matches by using method 2 relatively to method 1. Because of this constraint method 2 lags behind method 1, however after about 100 minutes they reach almost the same total number of matches. We choose to use the matches as proposed by method 2, because a bus is not likely to depart far before the planned departure time.

When we look at the share of consecutive stops for which the bus first departed the succeeding stop before departing at the current stop according to the trip planner trips assignment, we find that this method might not be the best. We plotted this number versus the sum of the absolute time differences in Figure 4.23. This figure shows that the percentage of consecutive stops with overlap rapidly increases and levels out at around 40 %. This is quite a large portion, especially when you take into account that these only include requests from group 1 (no bus trip has a travel time of over 120 minutes). If the match would really be wrong, this would mean that a preceding or succeeding bus trip of the same line is robbed from one or more matched requests. However, the fact that the request(s) get matched to the current trip tells us that the requests are closer to the current trip in the time dimensions. Other matching methods should be used to determine if this is the result of the matching method or because of the data. However, for this application we will have to accept these matches simply because we do not have reliable data by which we could separate the bad matches from the good ones.

For this research we choose to allow a sum of absolute differences up to 109 to include as much data as possible. From 109 minutes the line stagnates, a limit bigger as 109 would thus result in only a little more matches with the risks of including bad ones.

When we look at the type of requests that get matched using method 1 and method 2, regarding day, day-hour, aggregated hour, weekday, scheduledate, origin, destination, OD pair, travel time, request interval, line number and journey part number, see Appendix L, we see a similar distribution between method 1 and method 2. Only the characteristics based on hour really differ. Upon further investigating the night time/early morning hours get less matched using method 2.

Furthermore, we compared the number of requests we matched using method 2 with all requests based on the line number. For this, we first looked up to which line the requests belonged using the OD pair and the public line number (the public line number is not unique but including the OD pair makes it unique to a single line). This resulted in a percentage of requests for a line that are matched. The best matched line was *t810* with 92% of its 1481 requests matched. The least scoring line is *a042* with 14% of its 197 requests matched. Thus, the lines vary in total number of requests matched, but also in total number of requests. Based on these metrics we can easily separate smaller lines, which are more often served by buses without a board computer, from the regular ones. Therefore, we limit our study to the 20 lines with a matching score of 75% or higher with at least 10,000 requests, see also Appendix M.

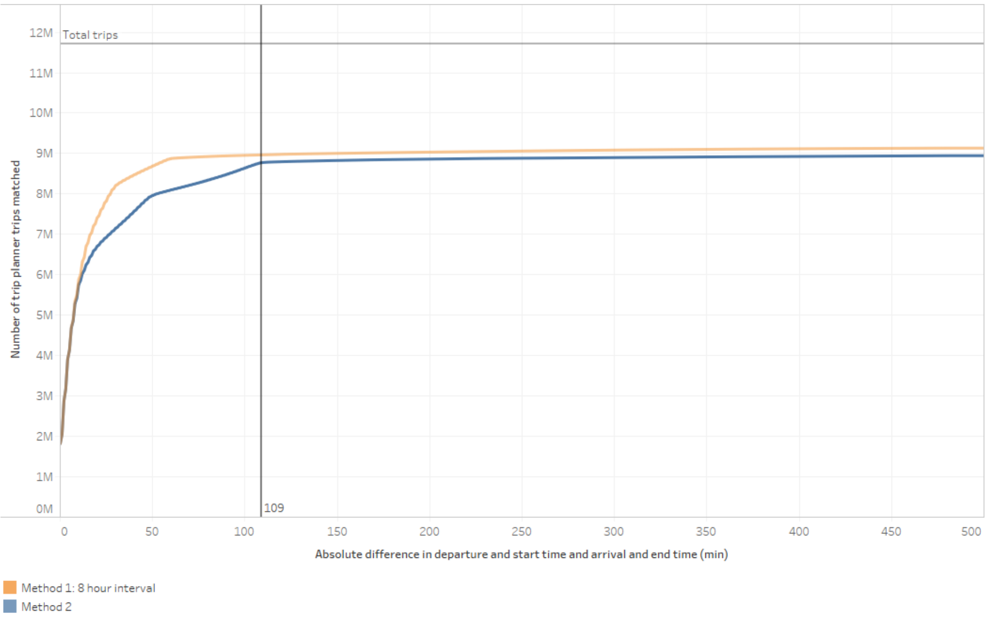


Figure 4.22: Number of trips matched versus the allowed difference between start and end time for method 1 and method 2.

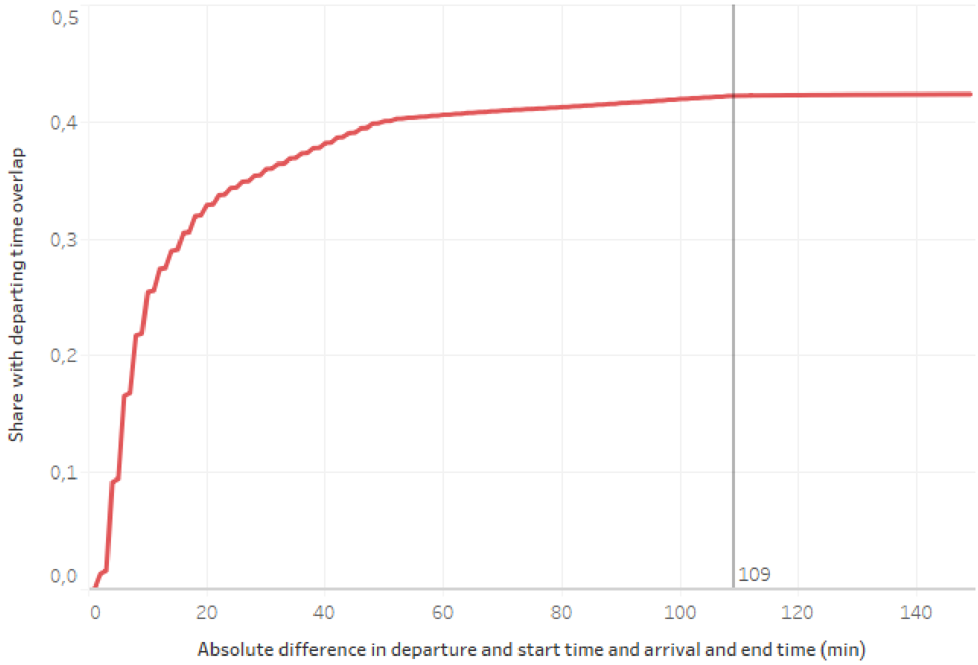


Figure 4.23: Departure time overlap according to the matched 9292 trips versus the sum of the absolute time differences . For the share of departure time overlap we only accounted for consecutive trips which both had 9292 trips matched to it with a request interval larger than 120 minutes.

4.6.3 Matching smart card data with bus data

This step describes the trip matching of the smart card dataset with the actual bus trips. Like the trip planner trips, we will match the smart card trips using the start location, end location, start time and end time, see Figure 4.19. Unlike matching the trip planner trips, we will not use a line number, since there is none available. The trip matching process is not an easy task for multiple reasons:

Different factors influence the punctuality of the bus, see Figure 4.20, therefore the recorded passage times often differ from the planned times. Furthermore, it could be that a passenger checks in after the bus has departed, especially if the bus driver has to make up for a delay and the passenger does not have the smart card at hand. In such cases the check-in time is later as the time of departure. The same applies to when a passenger checks out just before arriving at a stop.

It is not made easier by the determination of check in and check out stop by the bus equipment. Circular geofences with a radius of 15 meter are used to detect if a bus has arrived at a stop or if the bus has departed. When a bus leaves this geofence the current bus stop is already set to the next stop. So, it could be that if a person checks in when the bus already departed, the check in is registered for the next stop. And if a passenger checks out as soon as the bus departs, it could be that the check out is registered to the previous stop.

We will match the transaction trips with the actual bus trips using two methods. In both cases we will use the absolute time difference between the departure times and the arrival times as performance measure. For the bus times we will use the recorded variant where available, otherwise the target times are used. Using target times has consequences since in only 45% of the bus passages recorded in the dataset the bus is within one minute of the planned time, see Figure 4.2.

The first method is picking the best result from the bus trips which departs at the same stop and later arrives at the same stop with a target departure time that is within an interval of plus and minus 4 hours of the check in and end time. We will use this method as a baseline: it is not likely that a traveler checks in 4 hours before or after the bus departs a stop, however it is useful to validate the next method.

The second method also picks the best result from the bus trips which departs at the same stop and later arrives at the same stop. However instead of the 8 hour period, the check in and check out should have happened between the departure time of the previous stop and the departure at the next stop. The results of this method still need a constraint afterwards because the departure time at the previous stop and at the next stop is sometimes based on the planned time and sometimes on an arbitrary interval of 30 minutes instead of the time as recorded, see also paragraph 4.2.3. Therefore, a constraint is needed to limit the max time difference in order to reduce faulty matches and thus noise. We will determine this limit based on the elbow method.

In theory a check in time and a check out time after a passage is not possible since the bus would most likely have left the 15 meter geofence and the transaction would be recorded to the next stop. However, because both systems are independent and keep their own time, we will allow matches within a margin to allow for errors.

In case a transaction matches best with multiple bus trips we use the following order to pick the best:

1. check in time and check out time are both earlier as their respective bus departure time,
2. check in time is later and check out time is earlier as their respective bus departure time,
3. check in time is earlier and check out time is later as their respective bus departure time,

4. both check in time as check out time are later as their respective bus departure time.

The matching of the transaction is not biased towards the date, hour, hour type and date type as is shown in the table in Appendix LMatching trip planner trips to bus trips. However, some origins, destinations, OD pairs and travel times are less present after the matching. Furthermore, method 1 and method 2 show similar characteristics, where method 2 has the most matches. However, the quality of these matches is less assured because of the less constrained matching method. Thus, we will use the matches from method 2 with a maximum summed time difference of 10, as seen in Figure 4.24.

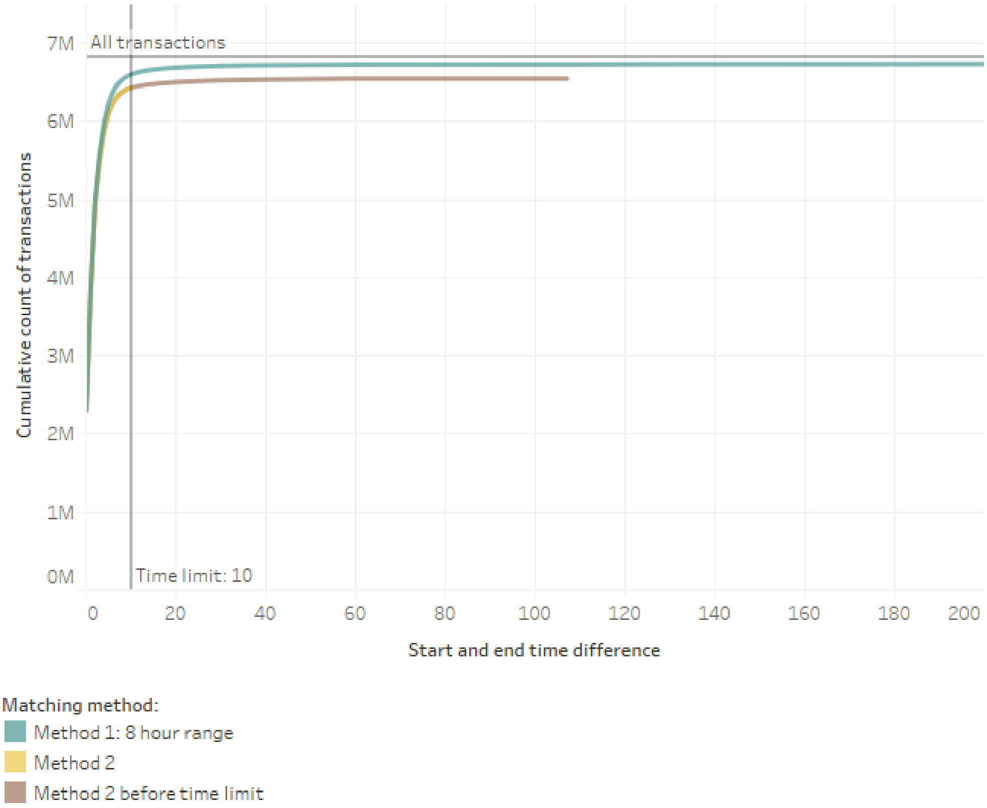


Figure 4.24: The number of transaction matched relative to the maximum summed absolute difference between departure time and check in time and arrival time and check out time

4.6.4 Matching weather data with bus data

This section elaborates the method of relating the weather stations to the stops. We will do this by giving the weather stations a normalized weight based on the inverse of the squared distance between a stop and the weather station (formula 4.1). The rainfall or rain duration at a given hour at a bus stop is then calculated as shown in formula 4.2.

$$a_{sw} = \frac{\frac{1}{{d_{sw}}^2}}{\sum_{w=1}^{8} \frac{1}{{d_{sw}}^2}} \qquad 4.1$$

$$R_{sh} = \sum_{w=1}^{8} a_{sw} R_{wh} \qquad 4.2$$

Where:

s denotes the bus stop

w denotes a weather station

h denotes the hour

a is the normalized weight of the relation between *s* and *w*

d is the Euclidean distance between *s* and *w* using the rd coordinates

R is the rainfall or rain duration

The inverse of the distance is taken to give a higher weight to the weather stations that are closer to the bus stop. The distance is squared to increase this effect. By normalizing the weights, we ensure that the individual weights are on the scale between 0.0 and 1.0 and the sum of the weights is equal to 1.0. An example for the computed weights is shown in Figure 4.16. In this figure the weights for *Groningen Hoofdstation* are calculated. With this method we construct a table with a record per stop and per hour with the amount and duration of precipitation. This results in a table with 5,805,525 records between 31-12-2016 23:00 and 01:00 on 15-04-2017.

4.7 Exploratory data analysis

In this section we will explore the combined dataset.

Figure 4.25 shows the number of stop passages, the number of people boarding and the number advices to board aggregated per hour of day partitioned by the type of schedule day. The bus data, smart card and trip planner lines show similar characteristics. There are some things that stand out: Only the weekdays have a morning and evening peak. In the weekends there seems to be a plateau between 10:00 and 21:00 hours. From the first subplot it can be noted that people are more eager to consult the morning journey as the evening journey. In the fourth plot there is an evening peak. Trips from the trip planner are most variable, whereas the bus data frequency in almost all instances stays the same.

The first subplot agrees with the findings from the literature study: The morning peak is sharper compared to the evening peak. The figure shows that the bus capacity follows the demand. Thus, there is additional capacity planned during the peak. Moreover, the figure shows that there are almost no buses during the night.

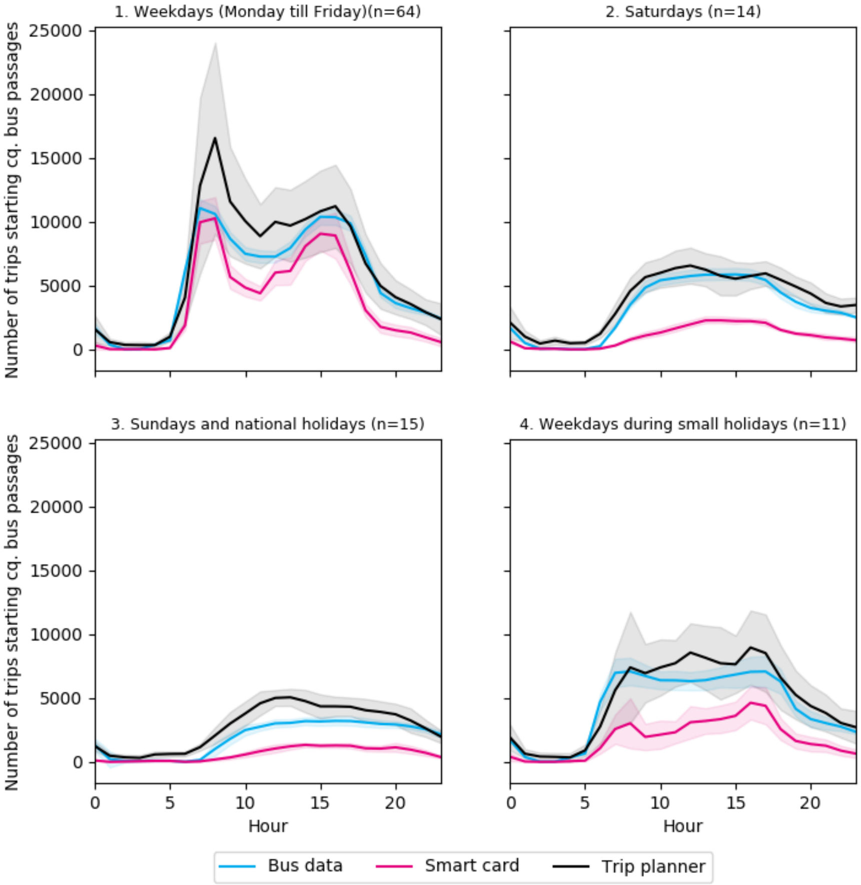


*Figure 4.25: The relative trip frequency starting in a given hour. The subplots are partitioned per schedule date type as is decribed above the subplot. The shaded area around the lines show the 2 * standard deviation interval.*

Figure 4.26 and Figure 4.27 show the ratio between the number of trip planner requests and the number of people checking in and out for the same setting. Again, the peaks are visible but now as dips: during the morning peak and evening peak relatively fewer people consult the trip planner. This could originate from different causes: more travelers are commuters who don't consult the trip planner for their everyday commute, the travelers are more efficient in consulting the trip planner and make their travel plan in less requests or the frequency of the buses during peak hours is sufficiently small (<10 every minutes) so people do not have to plan their trip in advance. It is interesting to note that on average on Saturdays relatively more requests per trip are made. Furthermore, for both type of days there is a higher ratio at the start and at the end of the day. This probably originates because these are the time periods the trip planner is consulted most frequent, there are less travelers and thus less people checking in and out, and there are no buses at night, thus the trip planner shows the last possible trip and the earliest possible trip when consulting a bus for within this time frame. The number of requests for the earliest and latest trip is thus not reliable.

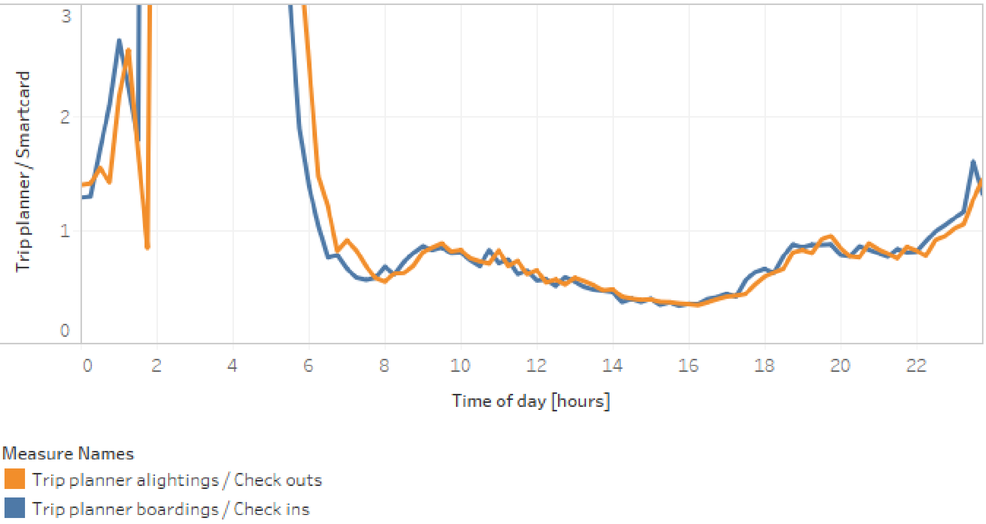


Figure 4.26: The number of requests per check in or check out on weekdays for lines which are available for analysis aggregated per 15 minutes

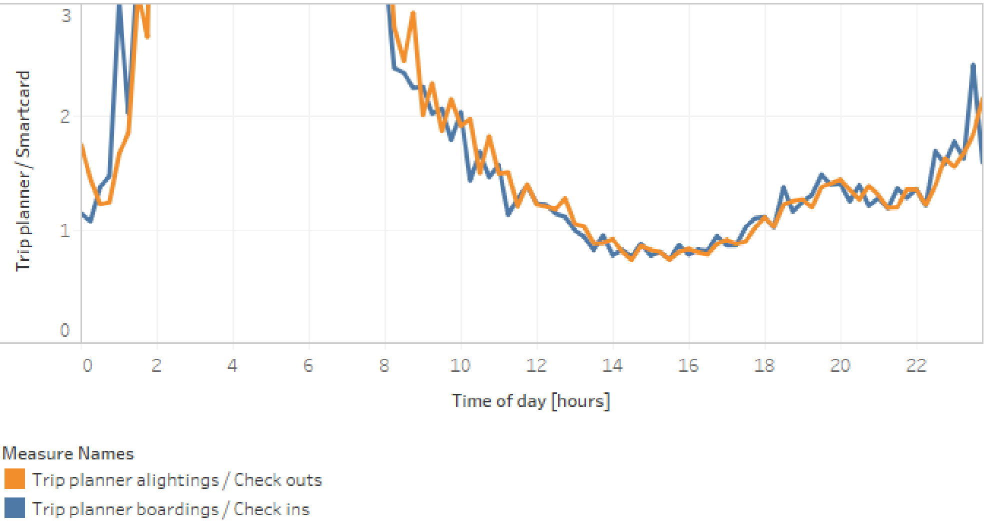


Figure 4.27: The number of requests per check in or check out on Saturdays for lines which are available for analysis aggregated per 15 minutes

4.8 Constructing the features

Because we have merged the datasets, we can now construct the features (synonym for attributes or variables) by calculating, gathering and collecting them from the individual datasets. The total number of constructed features can be found in Appendix N.

The first 21 features are extracted from the bus data dataset. Of these, 10 features (*id, no_historic_trips, operationdate, variant_id, Variant no, direction, lineplanningnumber, clustercode_9292, hour and scheduleday*) are purely to identify the records, to put the results into context and are not used for forecasting. Furthermore, the *variant id* (which incorporates *lineplanningnumber, variant_no* and *direction*) and *scheduleday* are used to partition the data, since these two features partition the data into groups with similar characteristics. The *variant id* encompasses data concerning the route and stops and the order of visiting the stops. Thus, the *variant id* gives an overall clue of the attractiveness of the boarding the bus at the same stop. This attractiveness may vary between the variants of the route since some variants visit more stops than others. The *scheduleday* is also used to partition the data since this feature separates the demand and the supply in clear patterns, see Figure 4.25.

Features 11 and 15 are included to incorporate the time dimension: *tijdsblok* and *weekday*. *Tijdsblok* divides the days into 5 bins: Saturdays are marked Saturday, Sundays are marked Sunday and during weekdays the days are divided in morning peak (5 till 10 am), evening peak (2 till 8 pm) and off peak. *Weekday* describes the day of the week, ranging from Monday to Sunday.

The *recordedpunctuality* (feature 12) describes the delay of the bus. This feature describes the influence of different processes: if there is a delay it is possible that people can board the bus who otherwise couldn't. However, if the delay is sufficiently large, other connections might be faster. This is especially true for a corridor with many buses.

Feature 13, *stopsleft*, and 14, *distanceleft*, are two features that describe the number of stops and the distance the bus still will service in the current trip. These features indicate the number of stops the passengers can travel to or how far they can go by boarding this bus and thus may tell something about the attractiveness of the current combination between the trip and stop.

Feature 16, *prev_headway_bin*, and 17, *next_headway_bin*, are relatively the headway between the previous and current bus and the headway between the current and succeeding bus of the same line. These features could be interesting, since if the headway is sufficiently small travelers are less bound to plan the trip in detail ahead of time and consult a trip planner. Travelers might even consider it more of a hustle to consult the trip planner than to just go to the bus stop and wait a few minutes. We defined these features as a categorical variable in order to be able to include the first and last trip of the day. Furthermore, we expect that the effect of the headway is also in steps: a headway of 7 or 10 minutes is expected to be not experienced different by travelers. Furthermore, there are some common values among the headways, see Figure 4.5. These common values each get their own bin. Since, the interval between the common values get bigger, the bins also increase in size. This is also in agreement with the expectation that the difference between a headway of 10 and 30 minutes is more extensional to travelers than a difference between 2 and 3 hours. In total there are 7 levels defined. These bins are shown in Figure 4.5 by the shaded areas.

Features 18, 19, 20 and 21 describe the number of buses arriving and departing in the 15 minutes prior and after. These features give an indication how many passengers might be able to transfer to the bus or alight to transfer to another bus. Furthermore, these features might give an indication of the popularity of the stop (and thus the overall demand at a stop).

From the smart card data we define 6 features. Features 26 and 27 are the *passenger_delta* and the *passenger_no*. *Passenge_delta* is the netto passenger flow at a stop passage (number of people boarding minus the number of people alightning). *Passenger_no* is the total number of passengers after the stop passage (*passenger_no* of previous stop plus *passenger_delta*) Both these features are not used for prediction but are later used to score the aggregated model. Feature 28, *cki_no*, is the dependent variable for the number of people boarding regression analysis and is defined as the number of people checking in at the stop passage. Feature 29, *cki_no_historic_avg*, is the average number of people boarding at the bus passage given the same stop passage in past weeks. We only include trips which have 4 or more weeks of data to reliably calculate the historic averages. We use the average over the median since the average is more constant and stabilizes more quickly. This feature is used as a baseline for the models. The next two features are the historic average of frequent travelers and the historic average of frequent transit users boarding at the stop passage. The difference between these two features is that one is based on the frequency of the smart card trip between the same origin and destination and during the same time period. The other uses the total number of trips the smart card is used for. Zhou et al. (2017) defined a regular traveler as a traveler with on average two trips per day. We will use a similar definition where a frequent traveler is defined as a traveler with at least 2 trips on average on working days. The smart card features *frequentie* and *total* are measured for the untrimmed dataset. The untrimmed dataset contains 32.6 weeks of which 3.2 weeks are school holidays. Therefore, we will count a smart card trip as belonging to a frequent traveler when *frequentie* has a value of 147 or higher and belonging to a frequent transit user when *total* has a value of 294 or higher. However, as Figure 4.15 shows, there are almost no frequent travelers when using a value of 147. Therefore, we will use the halve of this number (74) to define a frequent traveler, which means that travelers who board the same trip for half of the working days are considered frequent travelers. The number of frequent transit users is more loosely defined in regard to the number of frequent travelers, especially since a journey of a traveler can consist of multiple trips. These features are both included to see which one is better. Furthermore, schedule day 4 (and 2 and 3 during weekdays) was not defined as a separate time period in the smart card dataset. So, when building this model, the frequent transit users can be used, whereas the frequent traveler feature is expected to be less reliable since the holiday bus time table is used.

From the trip planner dataset we count the number of trips that start and end at the given stop passage. At first, we only take trips into account that have a request interval larger than or equal to 15 minutes because of one limitation: the included variables should be available at the time of the forecast. This means that when you want to predict the number of people boarding a bus 15 minutes in advance you can only include the trip planner requests that had a request interval on the journey part level of over 15 minutes.

Beside the trip planner features we will use other features as well to control for trends in the passenger demand. We will use the features as described in the literature review. However, we will not incorporate socio-economic or spatial/built environmental features as these only account for long term demand on an aggregated level. These features are mostly used for predicting demand when no historic data is available. Because we focus on the prediction of demand in the current system, we will use historic data instead to account for specific time-space demand characteristics. Also, the characteristics of other modes are not accounted for. In our case these modes are the train services, long distance buses and a few buses connecting other provinces with Groningen and Drenthe.

The total feature set can be found in Appendix N.

4.9 Conclusion

In this chapter we constructed a final dataset to use for the analysis. We investigated and cleaned the 4 separate datasets after which we merged them. We first aggregated the stops to clusters and matched these with the already existing clusters of the trip planner. Afterwards we matched the smart card trips and the trip planner trips to actual bus trips. This step involved estimating which bus trip was used based on time and space. We had to introduce some margin in order to get enough matches.

The trip planner data is matched to the bus trips based on departure time, arrival time (using the recorded or planned time depending on the time of requesting), boarding stop, alighting stop and bus line. We had to allow a total mismatch error of 109 minutes in order to match 75% of the trips.

The transaction data is matched to bus trips based on the best possible match concerning time difference between the check in and the bus departure time and the check out and the bus departure time. We thus don't exclude matches were the best matches lie after the bus has departed the stop, because of the unreliability of the recorded and planned times.

The weather data is matched to the stops by using a weighted sum. The weight of a weather station for a certain bus stop is based on the square of the inversed distance.

After merging the datasets, we augmented the final dataset by constructing different features from the data sources.

The current datasets had some challenges; Fusing the datasets from 9292, Translink and OV-bureau would be a lot easier if each dataset stores a bus trip id and uses the same set of bus stops. Furthermore, we think that a lot of noise in the data from 9292 can be prevented by incorporating an anonymous session identifier; an id which connects requests from the same user. Also, attributes like the browsing time would be helpful to determine the importance of a request.

301
820
9
30
153
730
553
529
837
116
70
674
815
376
645
371
968
566
546
389
245
184
89
967
398
83
412
659
835
841
901
653
10
243
405
506

5 Method

5 Method

In this chapter we will discuss the approach of investigating the forecasting potential of the trip planner usage data. Since we are trying to forecast a quantity, e.g. the number of passengers, we are dealing with a regression problem (James et al., 2013); The trip planner usage data is regressed towards smart card data. We will perform the regression analysis using machine learning techniques. Two separate models are trained; a model for forecasting the number of people boarding at a stop and a model for forecasting the number of people alighting at a stop. The forecasted number of people onboard are derived by combining the forecasts of these two models, see Figure 5.1. By first forecasting the number of people boarding and alighting, more information can be inferred.

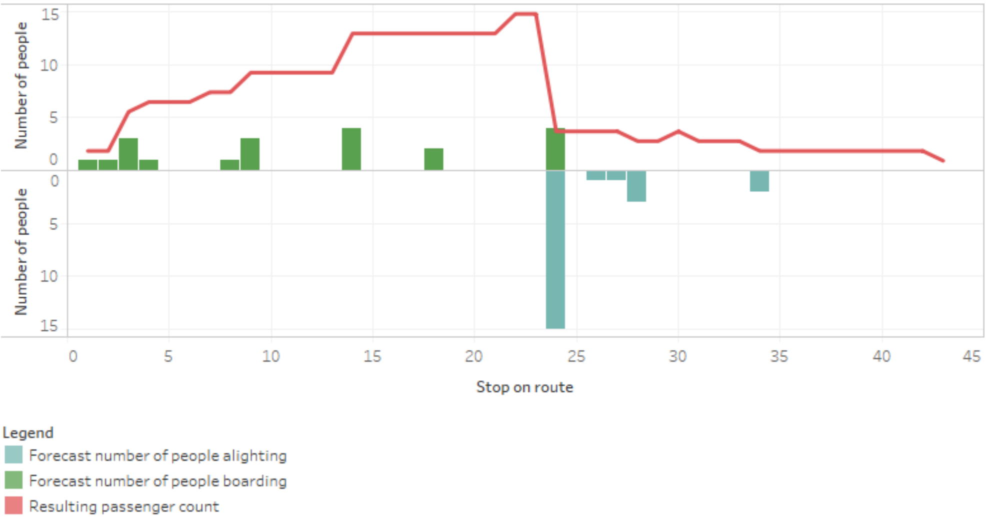


Figure 5.1: An example for the derivation of the number of passengers onboard from the forecasts of the two models.

Machine learning is a kind of artificial intelligence where the machine (computer) iteratively improves its performances in a certain task. The machine knows its improvement because each try is scored using a performance metric. Depending on the used model the machine tweaks certain parameters, for example weights, and tries again. This 'learning' process is repeated until a stopping criterion is reached or if the model cannot be further optimized. After the machine is trained it can be used as a forecasting model by entering the parameter values of the scenario for which the forecast is needed. The quality of the execution of this task depends on the training data, the form of this data, the used models and even the performance metric, see also Figure 5.2. These topics will be further discussed in this chapter.

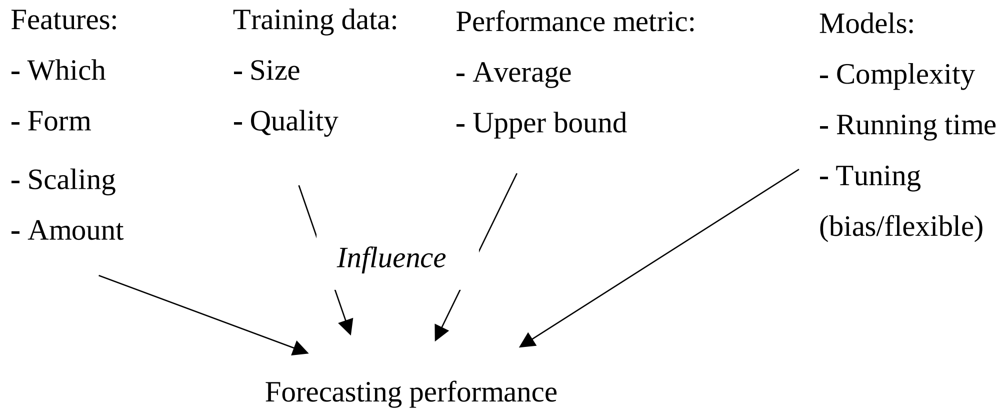


Figure 5.2: The influencers of the forecasting performance

We will implement our machine learning application in *Python*. *Python* has a great library for performing machine learning (including regression analysis) named *Scikit-learn* (Scikit-learn developers, 2018). We will use this library for the implementation of the different aspects discussed in this chapter.

Section 5.1 describes the used performance metrics, section 5.2 discusses which parts of the data we will use, sections 5.3 discusses the scaling of the features, section 5.4 describes which features are included, section 5.5 describes which models are chosen and section 5.6 describes how cross validation is used to assess how the model would perform to new unseen data.

5.1 Performance metrics

In this section we will discuss the metrics we are going to use to score the models. The choice of metrics is important since each metric has a trade-off. Improving the model on one metric could result in a decrease in the performance of another metric (Amrit et al., 2017).

For training and testing the model we are going to use 2 metrics: the root mean squared error (RMSE) and the R^2 (coefficient of determination). The RMSE is equal to the square root of the MSE, which is the most commonly used metric for regression models (James et al., 2013). Using the RMSE over the MSE increases the interpretability as the square root ensures that the RMSE is measured on the same scale as the dependent variable. Formula 5.1 shows the computation of the RMSE, where $\hat{y}_i$ is the predicted value and y_i is the true value for the ith record. A lower RMSE tells that the predictions are close to the true values. Because the prediction error is squared, the RMSE quickly increases when for some values there is a substantial prediction error. We will fit our models by minimizing the RMSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad 5.1$$

Another commonly used metric is the R^2. The R^2 measures the proportion of variance that is explained by the model and is defined by formula 5.2 (James et al., 2013), where $\hat{y}_i$ is the predicted value, y_i the true value and $\bar{y}_i$ the mean of the true values. The R^2 normally ranges between 0 and 1. If all the variance can be explained the R^2 will be 1. Since the R^2 is a proportion it can be used to compare the results with other research. The RMSE is not ideal for such comparisons because the RMSE is an absolute measure of the prediction error and thus is measured in units of y.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2} \qquad 5.2$$

Furthermore, we will score our final model with a metric introduced by Ohler et al. (2017): the percentage of correct predictions of passengers on a bus. This metric is defined on a different aggregation level than the RMSE and the R^2, since this metric uses both the predictions of the boarding as well as the alighting model. We will use the percentage of correct predictions metric to score our final model in a more real life perspective. Furthermore, we can benchmark our model with the model of Ohler et al. on this higher aggregation level.

We will also introduce a new metric: the percentage of correct peak load predictions. Unlike Ohler et al. (2017), we will make a difference between a positive and a negative error. Positive errors resemble an overestimation of the peak load in the bus. Negative errors denote an underestimation of the peak load. When operating on these forecasts, an underestimation of the peak load could lead to deploying smaller buses and, since the demand was underestimated, insufficient supply. In the same way overestimation could lead to overcapacity and thus higher costs. Deciding between a model that is biased towards overestimation or underestimation is thus equivalent to indirectly deciding between passenger satisfaction and costs. The current preference in the Netherlands is overcapacity, since traveler satisfaction is highly valued by policymakers.

5.2 Data selection

In chapter 4 the dataset was discussed. This dataset contains data from 20 different lines and 4 types of days. As stated, the demand and supply patterns change between the type of days. Therefore, we will partition the dataset by day. The demand and supply patterns also change with the lines and within the lines: the included lines have multiple configurations, see Figure 4.4. The attractiveness of a stop varies per line and, within the same bus line, per line configuration. Therefore, we will also partition the data per line configuration. Thus, for each line variant and each type of day we can train a model. Furthermore, it is likely that the travel behavior and trip planner usage behavior is dependent upon the service frequency. To investigate this, we will include a smaller partition with only trips with a headway of 10 minutes. As extra limitation we only use the bus trips that are operated between 8 AM and 8:14 AM so that we can analyze the influence of the morning peak. Initially we will start with the line configuration g554-1-0 (line g554, direction 1 and variant 0), because this line configuration has the most records in the peak, see Appendix O. G554-1-0 runs from Roden via the main train station of the city of Groningen to Beijum, see Figure 5.3 for the route and Figure 5.4 for the morning peak trips, has 43 stops with an average stop spacing of 631 meter (total route is 26 km) and takes about 61 minutes from begin to end. We will compare the models for this line configuration with models where all selected line configurations are included. In summary, we will train the models with 4 partitions of data. However, we will test the models with the same data partition to make the results comparable. We will use the trips of data partition 1 for testing the models.

1. line configuration g554-1-0 on workdays around 8 AM (1 line configuration, 239 trips and 10,277 records)
2. all line configurations on workdays around 8 AM (56 line configurations, 4173 trips and 138,694 records)
3. line configuration g554-1-0 for the total workday (1 line configuration, 2275 trips and 97,825 records)
4. all line configurations for the total workday (83 line configuration, 51,471 trips and 1,523,115 records)

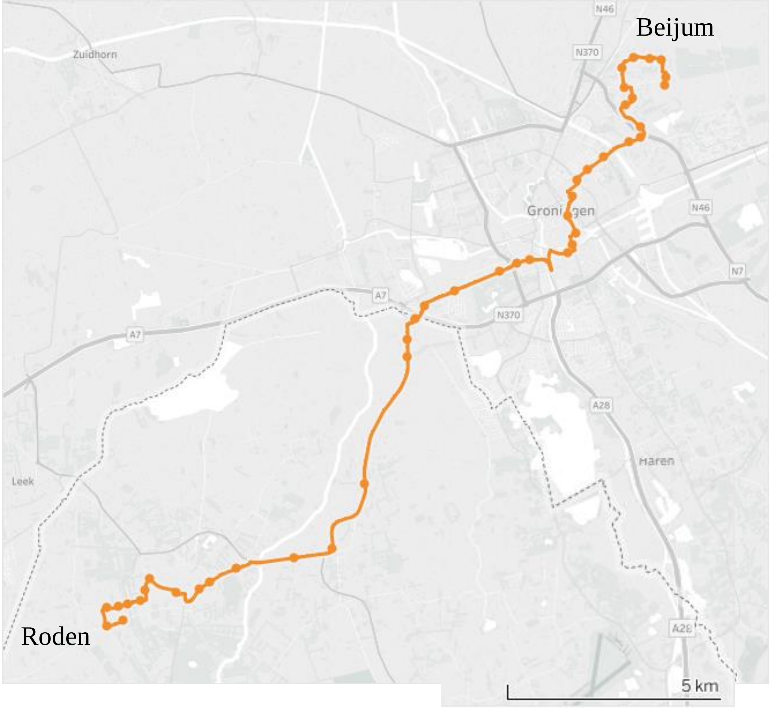


Figure 5.3: The line configuration g554-1-0 from Roden to Beijum with the corresponding stops

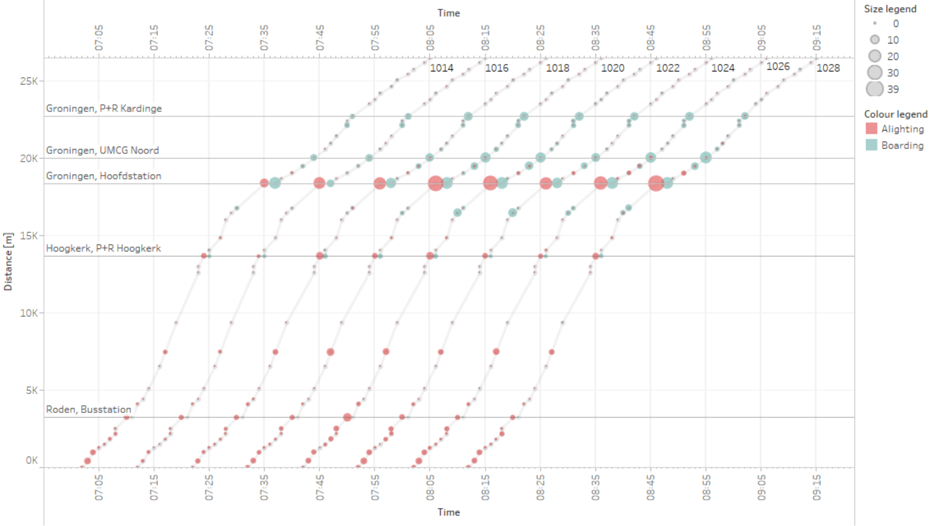


Figure 5.4: A distance-time diagram for the bus trips of data partition 1 for Monday 06-03-2017. The size of the circles denotes the historical average of the number of people boarding or alighting the stop.

5.3 Feature scaling

For many machine learning models it is important to scale the features before using them to train the model. Feature scaling removes the influence of the scale of the feature. There are two methods to scale the features: normalization and standardization. Normalizing adjusts the features to an interval between 0 and 1. Standardization transforms the features so that the transformed feature has mean 0 and a standard deviation of 1. We will implement standardization since this is less sensitive of noise.

5.4 Feature selection

Feature selection is an important step. If a variable is insignificant, it should be left out, otherwise the model becomes unnecessary complex and could overfit and thus has poor generalization purposes (Wei et al., 2014).

There are different ways to reduce the dimensionality. Filter methods use the characteristics of the data to select a subset. Wrapper methods feed the data in a machine learning model and uses the output metric to determine a good subset of features. Lastly, regularization and embedded methods penalize models with many features or prevent models to select too many (James et al., 2013). For instance, using random forests you can set the number of features the individual tree can choose from at each node and the number of nodes you allow. If you only allow a few nodes while choosing from many features some features are bound to be left out.

We used different methods to select the features. First, we used the Pearson's r statistic to filter the features. The Pearson's r is a normalized version of the covariance. Pearson's r tells us about the linear dependency between two variables. It is calculated by dividing the covariance by the product of the standard deviations, as can be seen in Formula 5.3.

$$r = \frac{\sum_{i=1}^{n}\left((x^{(i)} - \mu_x)(y^{(i)} - \mu_y)\right)}{\sqrt{\sum_{i=1}^{n}(x^{(i)} - \mu_x)^2}\sqrt{\sum_{i=1}^{n}\left(y^{(i)} - \mu_y\right)^2}} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \qquad 5.3$$

A perfect positive correlation has a Pearson's r of 1, a perfect negative correlation -1. Using a correlation matrix with Pearson's r we found many feature pairs having a high significant correlation, see Appendix P and Appendix Q. We therefore dropped as many features as possible. For instance, *rainduration*, *rainfall*, *prevrainduration* and *prevrainfall* all have a high significant correlation (Pearson's r > 0.69 and a p-value smaller as 0.01). We therefore choose only to keep one of those. In this case we will keep *rainduration*, since the Pearson's r with the dependent variable is similar for *rainduration* and *rainfall* (for both the case of alighting as for boarding) and it might better describe the impedance to travel during the whole hour. This way we reduce the number of features to 30. However, there are still a lot of features which have a low correlation (Pearson's r < 0.1) with the dependent variable. Thus, we will make a further selection of these 26 variables using wrapper methods. More specifically we will use the recursive feature elimination algorithm as implemented by *Scikit-learn* with a random forest as estimator.

5.5 Model selection

There are a lot of different models to choose from and there is no single model that works best in all scenarios (Raschka, 2015). Therefore, it is common practice to pick multiple models for comparison. Each model has its trade off and assumptions. Some models are good for prediction, others for inference. Some models are parametric, where the relationship is known beforehand and only the parameters have to be estimated, whereas others are more flexible at the cost of extra complexity. The most apparent tradeoff is the one between variance and bias. More flexible models have the risk of following the noise too close (overfitting), which results in bad generalization. Inflexible models on the other hand, may have a bias (to the mean) and may miss valuable patterns (Raschka, 2015).

Furthermore, it should be noted that there are two types of prediction error: reducible and irreducible error. The reducible error can be made smaller by improving the model, irreducible error cannot be improved upon since this is random and cannot be explained by the variables.

Like many studies from the literature review, we will include multiple linear regression models, support vector machines and neural networks. Furthermore, we will include decision trees and random forests as suggested by James et al. (2013). We will use the 5 models as implemented by *Sci-kit Learn*, these will be discussed in the following sections. Unlike some studies from the literature review we will not use time series analysis. Time series analysis has been implemented successfully to forecast public transport demand. However, it is not easy to incorporate external factors and only works on data that is aggregated over space and time. Since we explicitly want to investigate the predictive power of the trip planner data, e.g. external factors, and we want to make a forecast on a disaggregated level, we will leave time series analysis out of the scope.

5.5.1 Linear regression and multiple linear regression

The linear regression model is one of the most used models. This model forces a linear relation between the prediction variable *x* (or independent variable or regressor) and the outcome variable *y* (dependent variable or regressand), and thus only generates good results when a linear relation is to be expected. If multiple independent variables are used the model is called multiple linear regression (MLR).

By way of fitting data to the linear model, the optimal values for parameters *a* and *b* in formula 5.4 are found. These are the values that minimize the error rate. There are a lot of different methods to fit the data which can be linear and nonlinear. We will use ordinary least squares which fits the model by minimizing the summed squared difference between the predicted outcome and the observed one in the dataset. We will include an MLR model because these are easy to train and provide an upper bound.

$$y = ax + b \tag{5.4}$$

5.5.2 Decision tree

Decision trees are simple models and good for interpretation. A decision tree works as follows: Using a stepwise method the data is split in increasingly smaller branches. The output value for the branches is set to the mean of the true output of the samples in the branch. The decision rule used for splitting is based on one of the explanatory variables. For example: "The number of trip planner requests is bigger than 10." Furthermore, the decision rule which increases the chosen performance metric the most is selected. The used method is thus greedy, where the split is chosen which optimizes the current step instead of a split which might benefit the future tree.

When using a decision tree there is a risk of overfitting. This risk can be limited by setting a minimum number of samples in each end node of the tree and restricting the number of levels (depth) of the tree. By setting the minimum number of samples in an end node you prevent that each sample get its own branch. By limiting the depth, the final splits are cut off. These splits have a high chance to only contribute to overfitting. Because the decision tree uses a greedy approach, the model is highly dependent on the data it sees.

5.5.3 Random forests

Random forest for regression is an extension of the decision tree. Instead of fitting one tree, multiple trees are fitted. The trees each make a prediction. By taking the weighted average of these predictions the final prediction of the random forests is made. Unlike decision trees, these trees are trained on a resampled dataset using bootstrapping and by taking only a set of the features into account each split. Bootstrapping is a resampling method where you randomly pick a sample from the set with replacement. We will use the square root of the total number of predictors as the number of features to take into account each split as is typically done (James et al., 2013). Both methods ensure that the trees are more decorrelated. Because of the decorrelation, random forest is more stable and most often outperforms decision trees. However, this comes at a cost of reduced interpretability of the model.

The parameters to be tuned are: The number of trees, the maximum depth of the trees and the minimum number of samples of the end node. The number of trees should be sufficiently large to get a good result but should not be too large to prevent large training times. If the number of trees is large enough, a depth of 1 might also be sufficient. Setting the right minimum number of samples in the end nodes ensures a good variance-bias tradeoff.

5.5.4 SVR with radial basis kernel

Support vector regression is an extension to Support Vector Machines (SVM). SVM are classifiers that use a hyperplane to separate the classes. They maximize the margin around the hyperplane to make the model more robust. Only a subset of the samples is of influence on this hyperplane and margin, these samples are called the support vectors. Kernel functions can be used to model nonlinear relations efficiently when the separating plane is nonlinear. Kernel functions map a lower dimensional dataset into a higher dimensional feature space, and thus make the feature space nonlinear instead of the model. The SVM is extended towards the regression problem by introducing an ε-insensitive region around the hyperplane, called the ε-tube. The algorithm tries to find the tube that fits most datapoints. As with SVM, the datapoints outside the ε-tube influence the shape the most and thus are the support vectors (Awad & Khanna, 2015). Only residuals larger than a threshold, the ε, are taken into consideration. Support vector machines minimize the upper bound of the error instead of the mean (Ohler et al., 2017).

Two interesting kernels are the polynomial kernel and the radial kernel (James et al., 2013). We will implement an SVR with a radial kernel. For this model the following parameters must be tuned. The epsilon defines the size of the tube. If the residual is bigger as the epsilon the prediction is used in the loss function. C is a regularization parameter; a larger C gives more importance to minimize the error with the risk of overfitting. The size of γ influences the nonlinearity of the fit, where a bigger γ results in a more linear model. We will use a static gamma.

5.5.5 Neural networks

Neural networks are a name for algorithms which mimic the nerve cells of the brain: the output is calculated by propagating the input signal(s) through a network of neurons. Each neuron is linked to all neurons in the next layer. These links have weights. The neuron of the next layer calculates its value by entering the summed weighted inputs in an activation function. In the case of a regression analysis the last layer only has one neuron: the dependent variable. The number of (hidden) layers, the number of neurons in a cell, the learning parameter and the activation function are all subject to hyperparameter tuning. The weights can be derived using back propagation and stochastic gradient descent.

There are different types of neural networks, which mostly differ in structure. The most basic one is feedforward artificial neural network (ANN) as is described above. Other networks include recurrent neural networks (RNN) which includes lag variables by way of adding loops in the structure. We will incorporate an ANN model.

5.6 Cross validation

It is good practice to score your model with data that is not included in training the model, e.g. to use a dataset for training and a separate dataset for testing the model. This will give better insight in the generalization error. There are different methods to accomplish this. To keep as much data for training purposes, we will use k fold cross validation with 10 folds: the data will be split randomly in 10 parts with equal sizes, with these folds 10 models will be trained were for each model 9 folds will be used for training and the tenth fold for testing.

We are dealing with a sparse dataset where lots of dependent and independent variables are zero. For instance, depending on the line variant and schedule day type, 70 % or more (1:>2.3) of the stop passages have zero people boarding. This could result in a bias of the models towards predicting zero. This bias is unwanted, especially if this bias limits the models to forecast the unforeseen demand increases we are interested in. You could counter this effect by using over- or under-sampling. With this method you duplicate the values of the minority group or take a subset of the minority group. For instance, Amrit et al. (2017) used under-sampling in a classification problem to counter the bias. Before under-sampling the classes were present with a ratio of 1:20. In order to achieve a 1:1 ratio in the training dataset they split the minority class randomly in two groups and added an equal amount of randomly selected data from the majority class. To make the score more real-life representative, they used the original ratio again in the test dataset. They repeated this process 10 times in order to cross validate the score and increase the chance that each record was used once. We will incorporate a similar method. We introduce two auxiliary class variables: *cki_no_label* and *cko_no_label*. These features will be 0 if no travelers respectively checked in or out and 1 otherwise. We first split the data into 10 folds with in each fold the same ratio between the classes. Then we will randomly under sample the training data in order to get a ratio of 1:1 between the classes. The above is illustrated in Figure 5.5. We keep the ratio in the test data the same to mimic a real-life scenario.

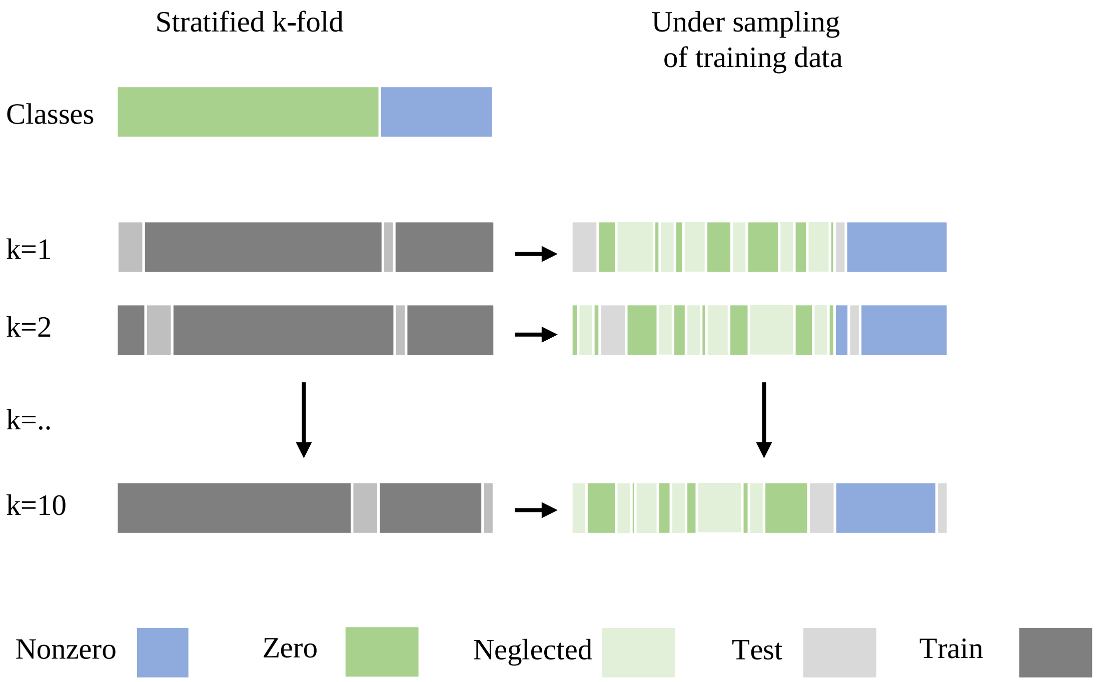


Figure 5.5: Cross validation set up

5.6.1 Pipeline

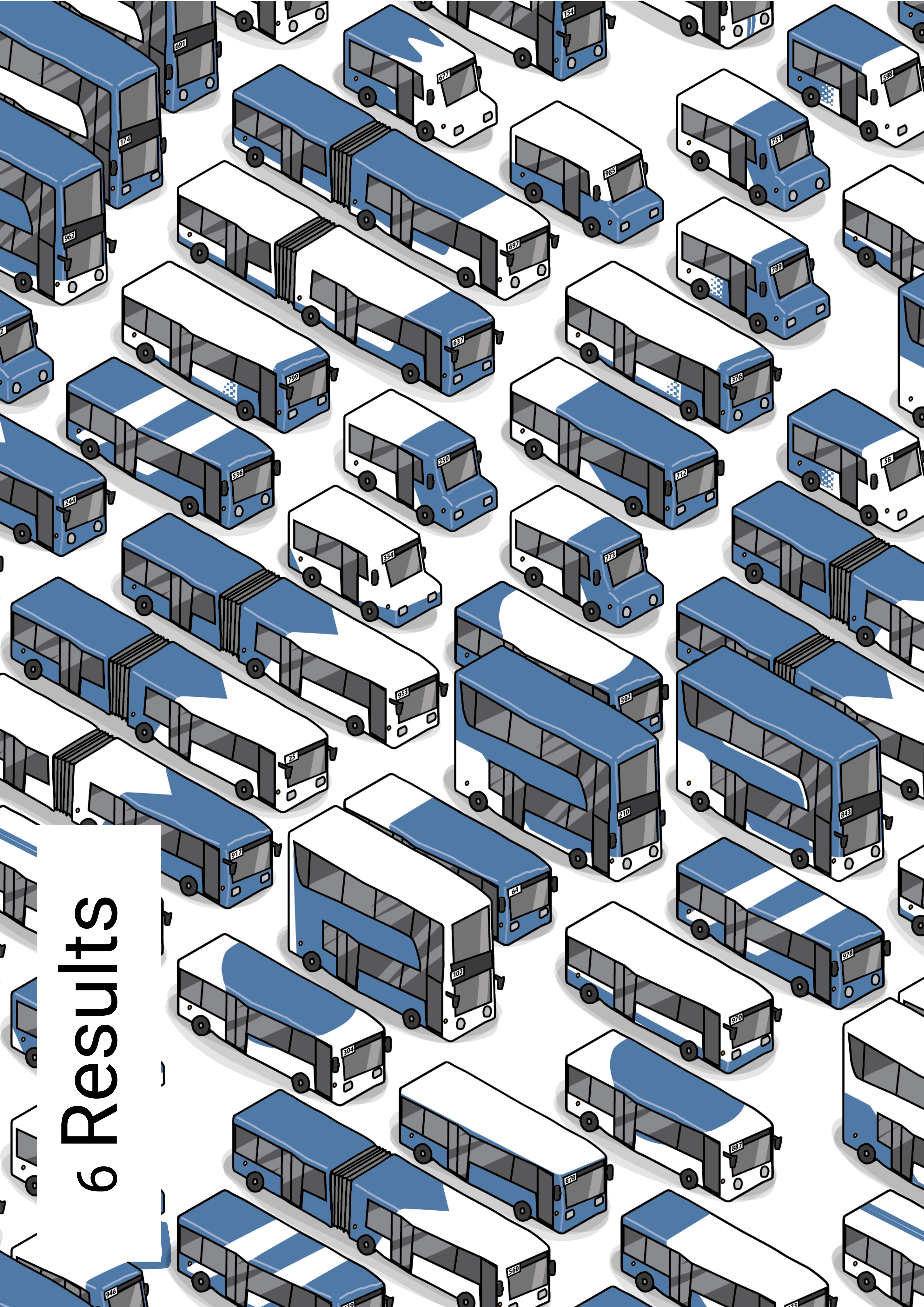
We implement the feature scaling, feature selection, hyperparameter tuning and the regression in a pipeline. For the hyperparameter tuning we will use *GridSearchCV* as implemented by *Scikit-learn*. By using cross validation, as described above, on this pipeline, we make sure that information on the test data does not slip to the model which would yield an over-optimistic score. Instead we scale and select the features, tune the parameters and train the model separate from the test fold.

5.7 Conclusion

We will not forecast the number of passengers directly. Instead we will train two separate models; one for forecasting the number of people boarding, the other for forecasting the number of people alighting. By combining the forecasts of these two separate models, we can derive the forecasted number of passengers.

We will optimize the machine learning models using the Root Mean Squared Error (RMSE). We constructed 4 different data partitions based on time and included bus lines. With these partitions we will investigate the effect of using extra, but less related, data on the performance. Furthermore, we will test 5 different machine learning models; multiple linear regression, decision tree, random forest, neural network and support vector regression. Through feature scaling and feature selection the feature set will be optimized for training data. We will validate the models using 10 fold cross validation and through the use of a pipeline. The pipeline makes sure that no information from the test dataset leaks to the training dataset.

6 Results


691
154
174
677
598
962
985
751
697
789
637
799
376
536
250
712
58
244
354
773
502
953
25
210
843
917
64
102
978
970
304
887
878
560
946

6 Results

This section will discuss the performances of the 5 models within the 4 defined data partitions discussed in the previous chapter. Each time we train the model with the whole data partition, but we test and score it only using the trips which are also present in data partition 1 (Workdays 8 AM for the g554-1-0 line configuration). This ensures that we measure the influence of using the data partition instead of measuring the performance of a different scenario.

For each model-data partition combination we found the best hyperparameters by hyperparameter tuning. We tuned the hyperparameters in two steps because of time constraints. We first used a more exhaustive grid for the *GridsearchCV* method (recall section 5.6.1) on partition 1. Afterwards we selected the best model per feature set size and used the parameters of these models to define a smaller subset of hyperparameters to use for the other partitions. After the first step the Neural Network model had still too many configurations for a reasonable running time. The hyperparameters *max epoch, alpha* and *tolerance* were reduced to one value. The new values where chosen to be in the middle of the 2 best values. The initial hyperparameters are given in Table 6.1, with in bold the final (hyper) parameter grid.

Model	Hyperparameters
MLR	Number of features included: **1**, **5**, **10**, **15**, **20**
DT	Number of features included: **1**, **5**, **10**, **15**, **20** Min samples in end note: 10, 20, 30, 40, **50**, 100 Max depth: **10**, 20, 30, 40, 50, 100

Model	Hyperparameters
RF	Number of features included: **1**, **5**, **10**, **15**, **20** Number of trees: 200, **500**, 1000, 2000 Min samples in end note: **10**, 20, 30, 40, 50, 100 Max depth: **10**, **20**, **30**, 40, 50, 100
NN	Number of features included: **1**, **5**, **10**, **15**, **20** Hidden layers: **(10,5)**, **(7,3)**, **(5,2)**, (2,2), (10,), **(5,)**, (2,) Max epoch: 1000, 2000, **1500*** Alpha: 0.01, 0.001, **0.0005*** Tolerance: 0.01, 0.001, 0.0001, **0.0005*** *Included after initial parameter set.
SVR	Number of features included: **1**, **5**, **10**, **15**, **20** C: 0.1, **1**, **10**, 100, 1000 Epsilon: 0.01, **0.1**, 0.5, 1, 2

Table 6.1: Initial hyperparameters per model. In bold the hyper parameters as used for the remainder of the analysis.

To put the performance of the five forecasting models in perspective, we will also use two heuristics for forecasting. The first heuristic is currently used by OV-bureau to plan the reinforcement buses. This heuristic predicts that the same number of people as the week before will board (or alight). The second heuristic outputs the historic average over the previous weeks of the number of people boarding (or alighting) and thus smooths outliers. This heuristic can be interpreted as predicting the number of people boarding (or alighting) on a typical day.

6.1 Important features

Before we get to the model results, we will analyze which features are important. We will use the feature importance rating provided by the *Random Forests Regressor* of *Scikit-Learn*. The importance of a feature will vary per model type, number of features included and data partition. Therefore, these scores only give an indication. Figure 6.1, Figure 6.2, Figure V.1 and Figure V.2 in Appendix V show the feature importance for the boarding and alighting model for the data partitions *g554-1-0/workday* and *g554-1-0/Workday 8 AM.*

As expected, the most important feature is the historical average of the number of people boarding (alighting) a certain stop for a specific trip. This feature shows to be a good baseline for forecasting. The number of frequent travelers is also among the top features. Moreover, the feature that is constructed based on the total trips made by the smart card outperforms the feature that is based on the number of trips during the same time period and between the same origin and destination.

The best performing features based on the trip planner data are the historic average and the number of requests aggregated over a larger period of time. The historic average is expected to serve as an indicator of the popularity of the stop-time-route combination. It could also be that the historic average is extra useful in combination with the number of requests (*start_15_total* and *end_15_total*). However, as indicated above, the number of requests feature is less valuable as the number of requests aggregated over a larger period of time, which could be explained by the fact that people are taking a similar bus, but not the exact same bus as stated in the travel advice. The number of requests in the previous trip and in the next trip are also important. More so than the number of requests for the previous stop or the number of requests for the next stop. The number of buses arriving in the previous 15 minutes, the number of stops left and some of the headway features are also deemed important. The feature *before_buses_arrival* seems to be more important for the boarding model than for the alighting model, the inverse is true for the feature pair *cki_frequent_traveler_same_corridor_historic_avg* and cko*_frequent_traveler_same_corridor_historic_avg*.

Features that were deemed to be unimportant are all *weekdays*, *rainfall* and most *hours*. In Figure 6.1 and Figure 6.2 it looks like that only the hours are included where there are demand peaks (7 and 8) or valleys (1, 5, 6, 23 and 0). The *rainfall* feature does not seem to be important. We expected that this feature would have an effect since there would be a mode choice change. There are multiple possible reasons why this feature shows no effect; It could be that there are insufficient rainy days in the study for the feature to have an importance. It could be that the mode change between active (walking and cycling) and public transport is almost equal to public mode to private (car). It could be that the chosen lines do not run along a corridor which are also used by active mode users, so there is no mode choice change. Or this could be because weather has more impact on the first leg of the journey; while on trip, bad weather has less influence since the traveler is already out and about and has fewer options. Or it could be, because the effect of rain is already captured in the trip planner request statistics. The *weekday* feature also has no importance. Thus, there is no clear difference between the weekdays. However, it could also be that the difference between the weekdays is captured better by other features, like the *historic averages*.

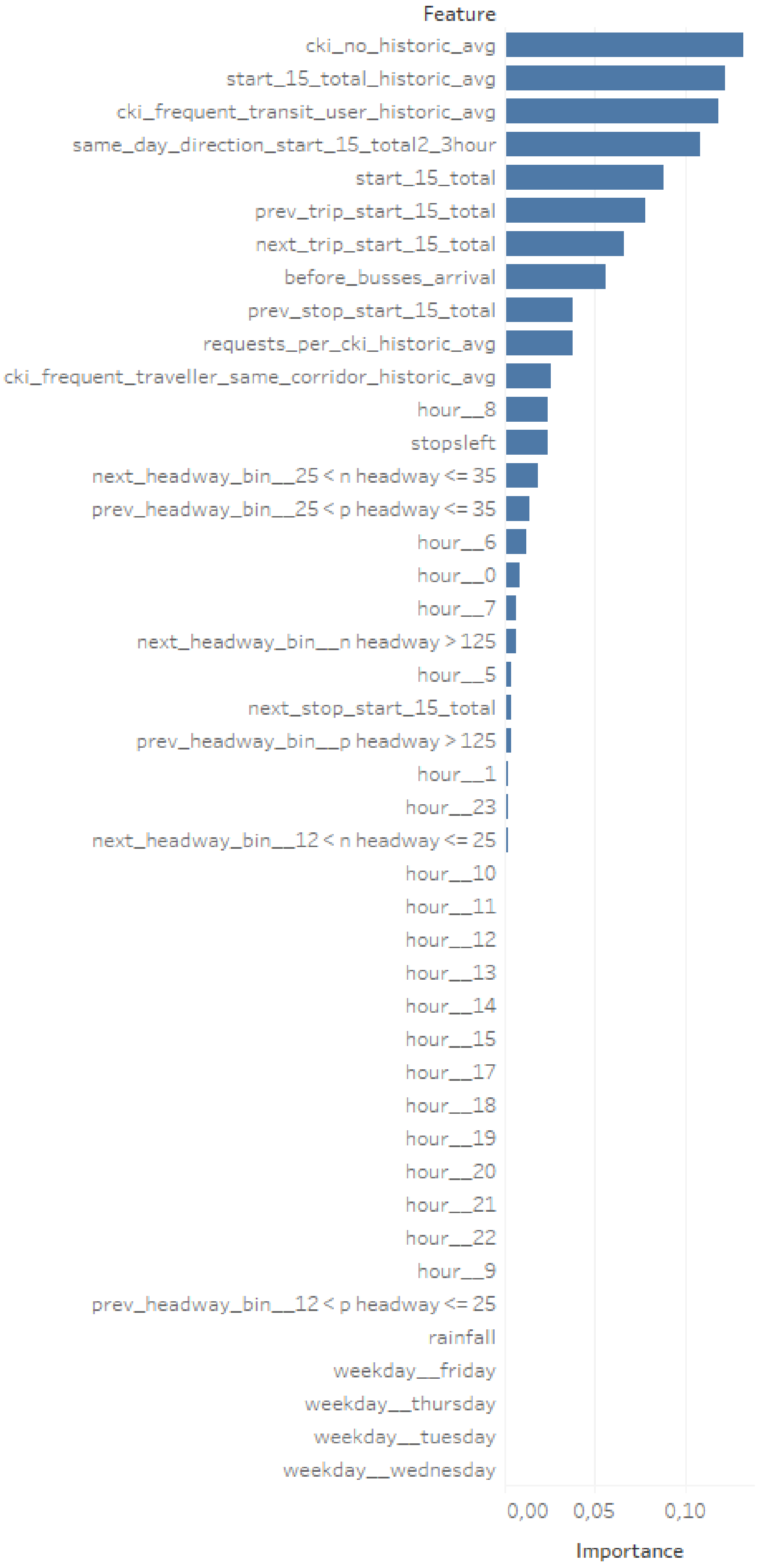


Figure 6.1: Feature importance of the boarding model for the line variant g554-1-0 and during a workday

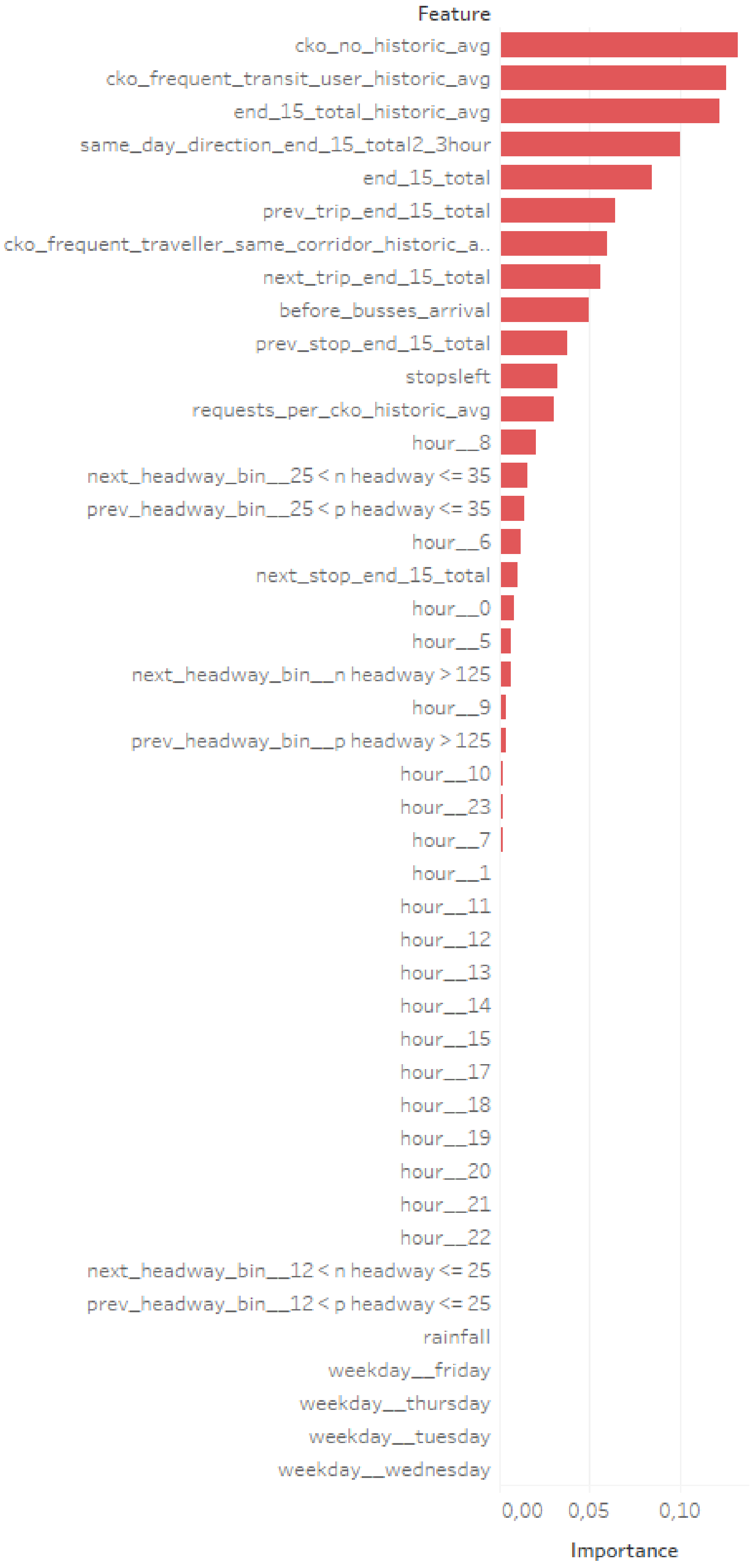


Figure 6.2: Feature importance of the alighting model for the line variant g554-1-0 and during a workday

6.2 Predicting the number of people boarding

Figure 6.4 shows the performance of the best boarding models per partition of data and number of selected features. The figure shows that the best R^2 score does not always coincide with the best RMSE; a better R^2 does not always lead to an improvement in RMSE. This could be because the R^2 score refers to the average error, whereas RMSE penalizes larger errors more. The best results per model, the models with the lowest RMSE, are summarized in Table 6.2. The best performing model is a Random Forest with 500 trees with each a max depth of 20 and a minimum of 10 samples in the end notes. This model has an RMSE of 1.632 and a R^2 of 0.722. It should be noted that the models Neural Network, Decision Tree and Multiple Linear Regression are not far off. The Support Vector Machines for Regression performs the worst of the five models, which makes this model unattractive, even more so because of the larger training times (more than 10 times as long).

Heuristic 2 performs slightly better as SVR, whereas Heuristic 1 has the worst score of 2.107. Thus, all investigated models perform better as the current practice in terms of RMSE of predicting the number of people boarding.

Figure 6.3 shows the performance of all models and partitions for only the trips of the partition 1, *g544-1-0/Workday 8 AM*. The RF model with the partition *g544-1-0/Workday 8 AM* has the lowest RMSE with a value of 2.55. Overall the RMSE for these trips is higher as the best performing scores as presented before. Apparently, the trips outside the morning peak are easier to predict which brings the overall RMSE down. Thus, for the morning peak it helps to use only data of the morning peak.

When looking at the RMSE per stop cluster in Figure W.1 (Appendix W), we can see that some stop clusters have a significantly higher RMSE. The biggest RMSE is scored by the stop cluster *Groningen, Hoofdstation*. Furthermore, it seems that there is almost a perfect correlation between the standard deviation and the RMSE.

Unfortunately, the subsampling had no effect. In fact, Figure X.1 (Appendix X) shows a small loss because of the subsampling.

We can conclude the following; subsampling has a slight negative effect and can be neglected for further development. Trips outside the morning peak are easier to predict. You get the best result when only including data of the same scenario, in this case when only including data from the same line and time period.

Some stops are harder to predict than others. We might improve this by including other stop characteristics. Furthermore, the random forest model seems the most promising, this model uses the maximum number of features allowed. The performance might be improved by incorporating more features and allowing for a larger feature set size.

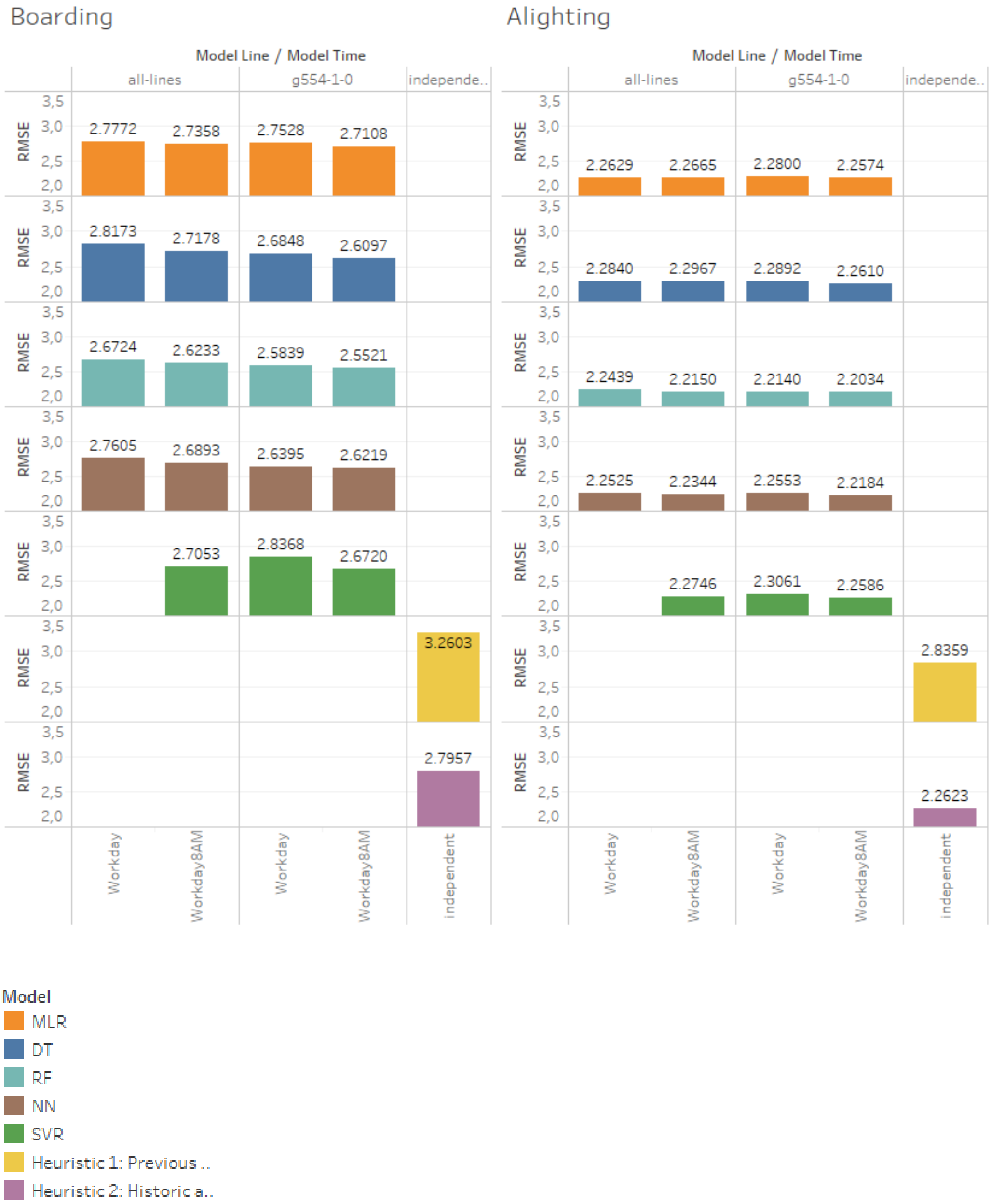
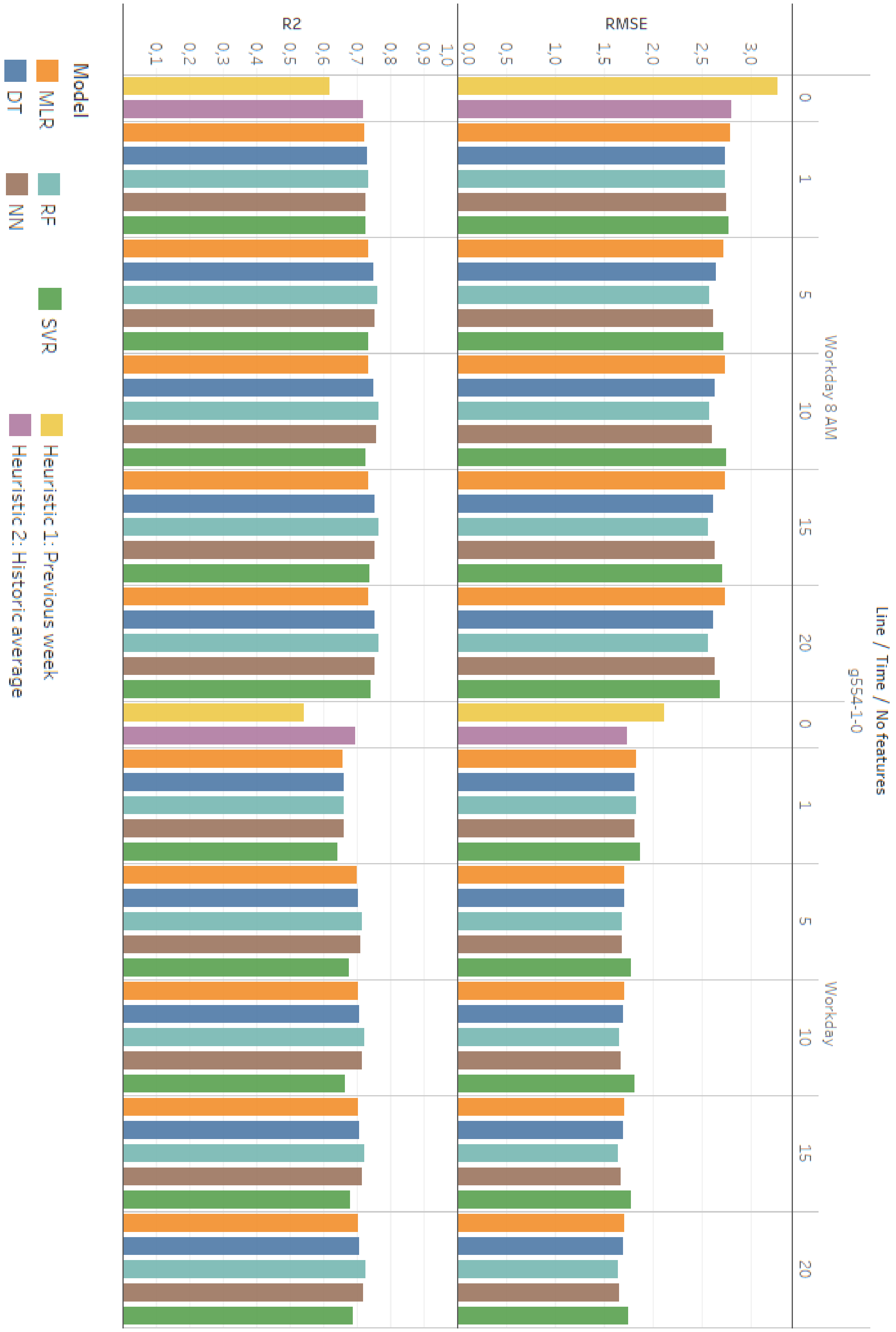


Figure 6.3: The RMSE of predicting the number of people boarding for trips of line variant g554-1-0 around 8 AM when first training the model with the line and time partition as stated.


RMSE
0,0
0,5
1,0
1,5
2,0
2,5
3,0
R2
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,8
0,9
1,0
Line / Time / No features
g554-1-0
Workday 8 AM
Workday
0
1
5
10
15
20
Model
MLR
DT
RF
NN
SVR
Heuristic 1: Previous week
Heuristic 2: Historic average

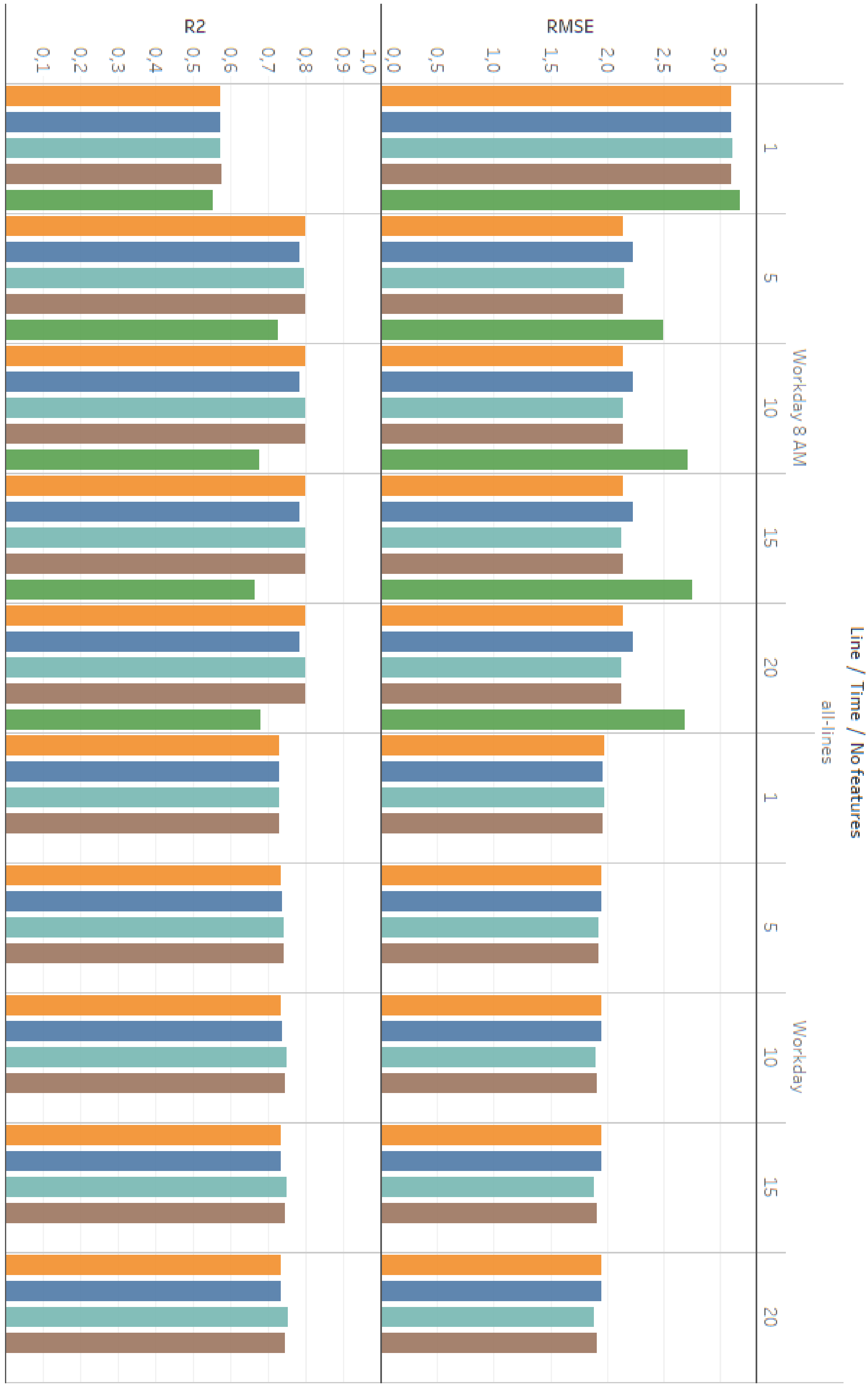


Figure 6.4: The RMSE and the R² for the prediction of number of people boarding using the 4 partitions (left page g554-1-0, right page all 19 lines), a number of features of [1, 5, 10, 15, 20] and 5 machine learning models.

Model	Partition	No features	RMSE	R^2	Fit time	Hyper parameters
MLR	Workday / g554-1-0	15	1.693	0.700	115	
DT	Workday / g554-1-0	15	1.679	0.705	121	Min samples in end note: 50, Max depth: 10
RF	Workday / g554-1-0	20	**1.632**	0.722	147	Number of trees: 500, Max depth: 20, Min samples in end note: 10
NN	Workday / g554-1-0	20	1.651	0.715	149	Alpha: 0.0005, Hidden layers: (10,5), Max epoch: 1500, Tolerance: 0.0005
SVR	Workday / g554-1-0	20	1.743	0.683	1737	C: 10, Epsilon: 0.1
Heuristic 1: Previous week	Workday / g554-1-0	-	2.107	0.539	-	
Heuristic 2: Historic average	Workday / g554-1-0	-	1.722	0.692	-	

Table 6.2: Best results per model for predicting the number of check ins. Training and testing on the same partition. Bold: global optimum.

6.3 Predicting the number of people alighting

Figure 6.5 shows the performance of the best models per partition of data and number of selected features. The best results per model are summarized in Table 6.3.

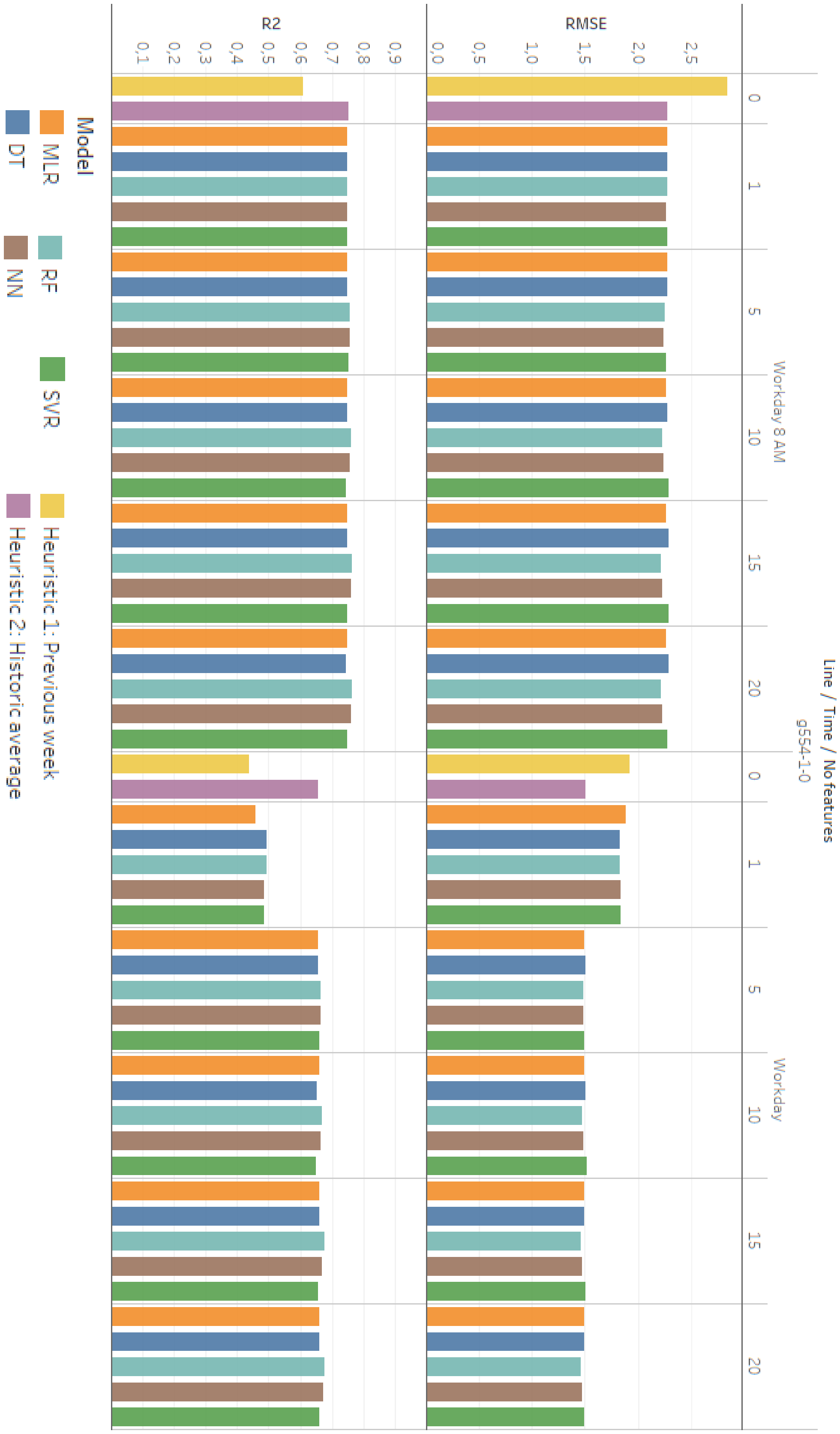
Like the boarding scenario, the RF model in the g554-1-0/workday partition has the lowest RMSE. This model has 1500 more trees which have 10 more levels as the best RF model in the boarding scenario. The overall RMSE scores are lower, however, the R^2 are also lower compared to the boarding scenario. This means that overall slightly less variance is explained, although there are less big predictions errors and substantially more small errors as in the boarding scenario.

Model	Partition	Number of features	RMSE	R^2	Fit time	Hyper parameters
MLR	Workday / g554-1-0	20	1.483	0.658	85	

Model	Partition	Number of features	RMSE	R^2	Fit time	Hyper parameters
DT	Workday / g554-1-0	15	1.485	0.656	130	Min samples in end note: 50, Max depth: 10
RF	Workday / g554-1-0	20	**1.448**	0.674	207	Number of trees: 2000, Max depth: 30, Min samples in end note: 10
NN	Workday / g554-1-0	20	1.461	0.668	137	Alpha: 0.0005, Hidden layers: (7,3), Max epoch: 1500, Tolerance: 0.0005
SVR	Workday / g554-1-0	5	1.482	0.658	923	C: 10, Epsilon: 0.1
Heuristic 1: Previous week	Workday / g554-1-0	-	1.908	0.434	-	
Heuristic 2: Historic average	Workday / g554-1-0	-	1.495	0.653	-	

Table 6.3: Best results per model for predicting the number of check outs.

Similar to the boarding model, the model improves by using only data from the same partition, see Figure 6.3. Random forest again scores the best performance for forecasts of trips of data partition one with an RMSE of 2.20. Overall the RMSE for the alighting model is lower as the RMSE for the boarding model. This suggests that the number of people alighting is more predictable than the number of people boarding.


Line / Time / No features
g554-1-0
Workday 8 AM
Workday
0
1
5
10
15
20
0
1
5
10
15
20
RMSE
0,0
0,5
1,0
1,5
2,0
2,5
R2
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,8
0,9
Model
MLR
DT
RF
NN
SVR
Heuristic 1: Previous week
Heuristic 2: Historic average

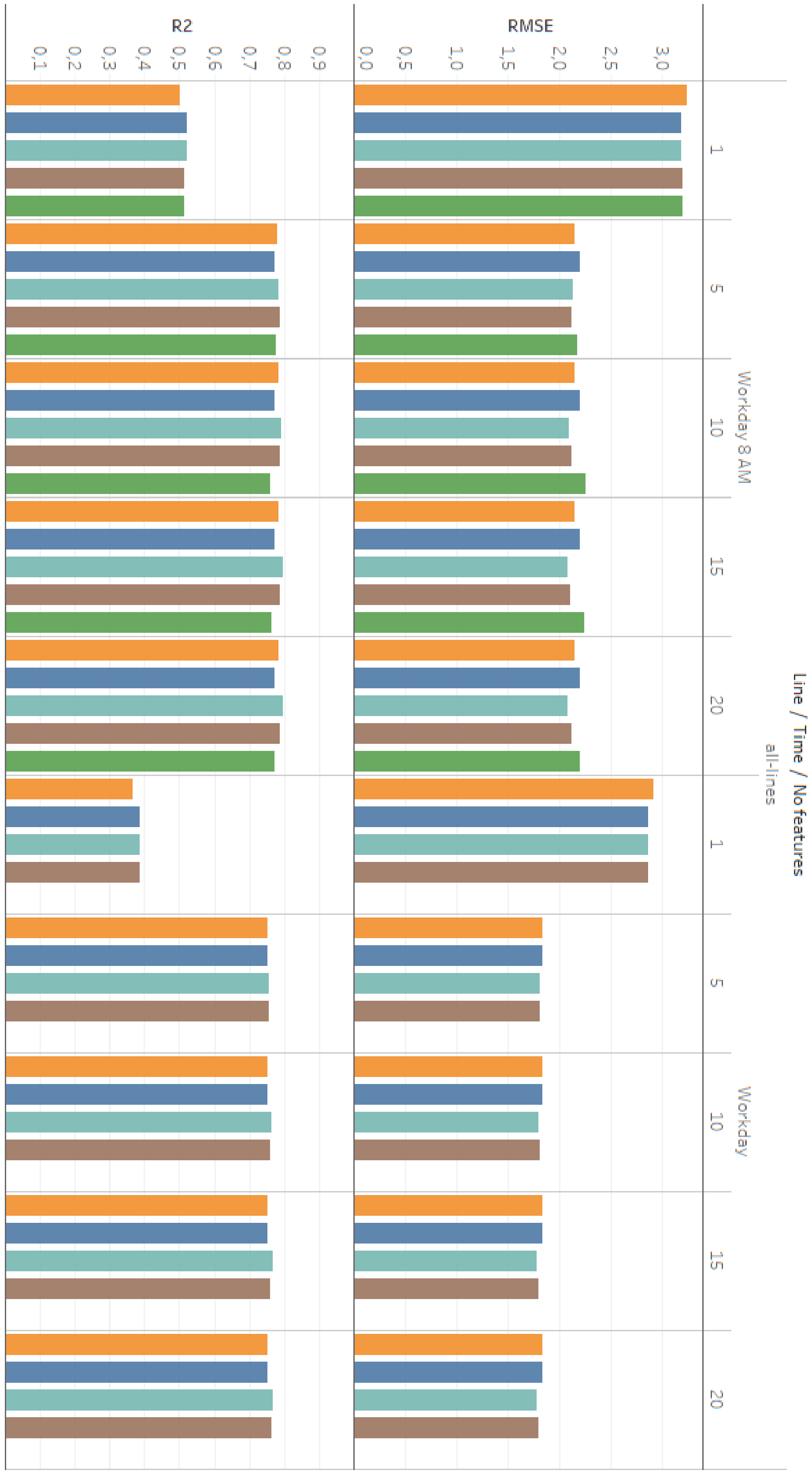


Figure 6.5: The RMSE and the R² for the prediction of number of people alighting using the 4 partitions (left page g554-1-0, right page all 19 lines), a number of features of [1, 5, 10, 15, 20] and 5 machine learning models.

6.4 Predicting the number of passengers

The number of people boarding and alighting predictions of the best performing models per type of machine learning model and per data partition, are used to calculate the predicted number of passengers on board after a stop, see Figure 6.7. The used formula is shown in 6.1.

$$P_s = \sum_{i=0}^{s} B_i - \sum_{i=0}^{s} A_i \qquad 6.1$$

In formula 6.1, P_s denotes the predicted number of passengers onboard after stop *s*, *i=0* denotes the starting stop of the trip, *s* denotes the current stop, B_i and A_i denote the number of passengers boarding and alighting stop *i* respectively.

Figure 6.6 shows the RMSE for predicting the number of passengers onboard. The results in this figure are shown per data partition used for training and per machine learning model. The best performing models of each machine learning model regarding the boarding and alighting models are used to come up with the prediction for the number of passengers. You could also match a boarding model of one type with the alighting model of another. This could be especially interesting since the boarding model and the alighting model seem to be different in terms of predictability and important features. However, this is currently left out of the scope. Figure 6.6 also shows a second type of prediction. This prediction first rounds the prediction of the number of people boarding and alighting to nonnegative integers before summing them like in formula 6.1. The lowest RMSE is reached by the predictions of heuristic 2 (directly using the historical average of the number of people boarding and alighting a stop) which has a RMSE of 8.603. The succeeding best performing models are RF when using unaltered inputs and MLR when first rounding the inputs. These models have a RMSE of 8.718 and 8.774 respectively. Both these models perform best in data partition 2 (*all-lines/workday 8 AM*).

Thus, a better performance in both the boarding and alighting models does not automatically mean the best performance for the number of passengers when combining the two. Not only does the best type of model change (from RF for the boarding and alighting model to Heuristic 2) but the optimal data partition to feed the model changes also (from partition 1 to partition 2). Furthermore, the overall RMSE increases a lot between the passenger model and the boarding and alighting model. This could partly be because the boarding and alighting models do not work together very well. However, it could also be that the prediction errors get accumulated along the trip to some extent. Thus, it seems that the proposed method for forecasting the number of passengers is not viable.

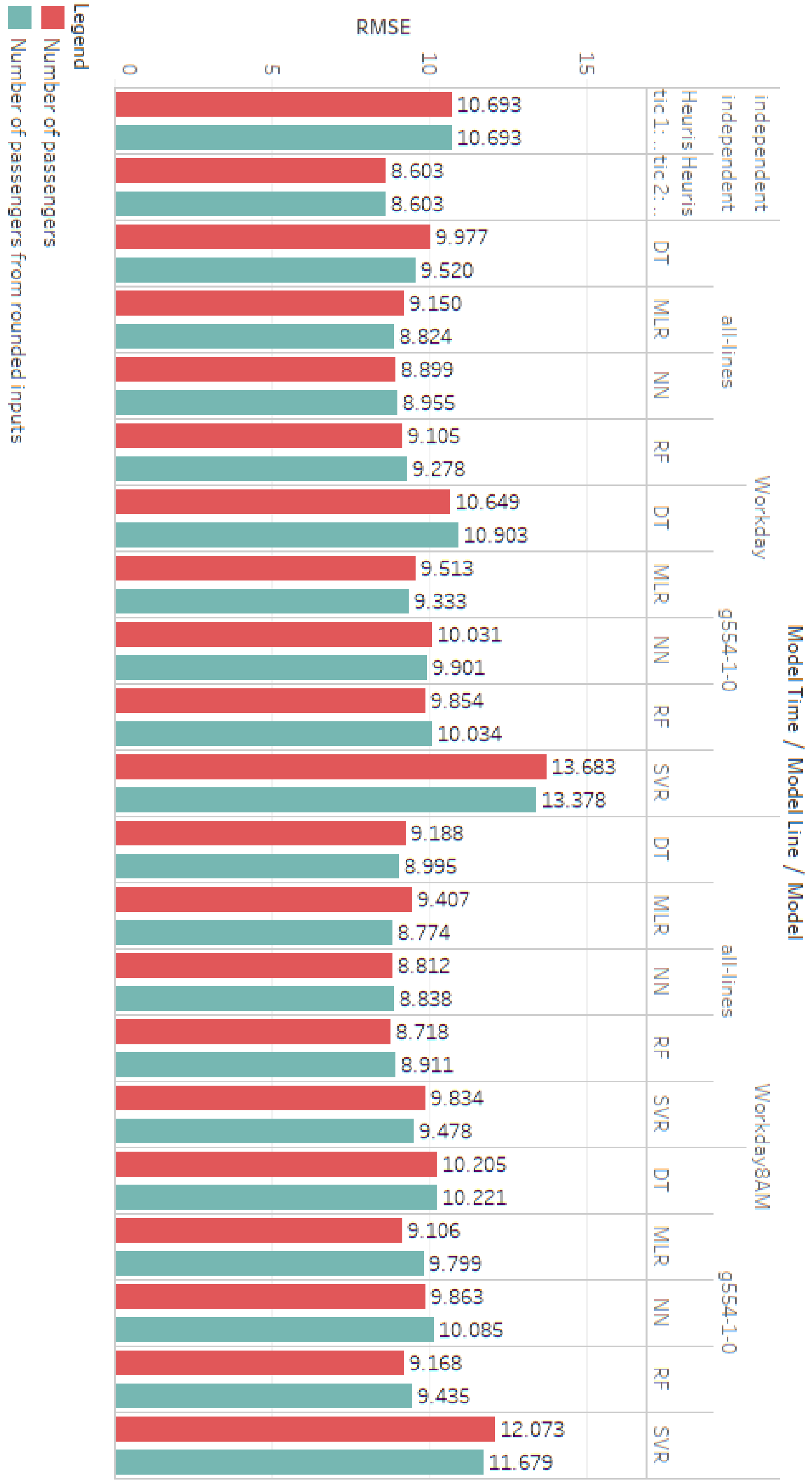


Figure 6.6: The RMSE for predicting the number of passengers for g554-1-0 trips during morning peak given a data partition and model. The passenger predictions are made by counting from the start the number of people boarding and alighting the bus at each stop. For the red predictions the boarding and alighting predictions are first rounded to nonnegative integers.

We will investigate the results of the passengers models Heuristic 2, RF, rounded MLR (all using data partition 2) further using the custom metrics as discussed in section 5.1: the percentage of correct predictions for each stop as well as for the max load during the trip. We will also include the Heuristic 1 model for reference.

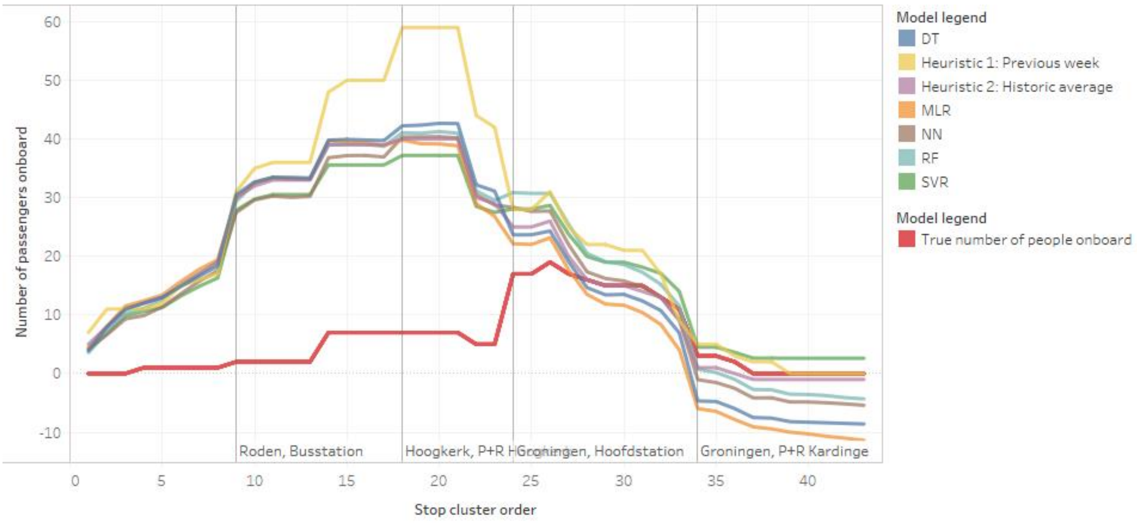


Figure 6.7: The predictions for the number of people onboard versus the real number for trip 1022 of line variant g554-1-0 on 15-02-2017. In this case the models overestimate the max load.

Figure 6.8 shows the percentage of predictions with an absolute error within a certain tolerance level. The RF model has 58,9% of the predictions correct within a tolerance of 5 passengers and 84.08% within a tolerance of 10 passengers. This is a better performance as reported by Ohler et al. (2017). The Heuristic 2 is better with 63.02% and 84.56% relatively. At the lower tolerance levels Heuristic 1 outperforms the RF model. However, from a tolerance level of 6 onwards, Heuristic 1 performs worse. Thus on average, heuristic 2 performs better as the RF model. However, it could be that these models perform better given a specific scenario. Further investigation is therefore needed to determine if the RF model outperforms Heuristic 2 for trips of a specific scenario, e.g. trips with a sudden unexpected demand peak.

Figure 6.9 shows an histogram of the percentage of correct max load prediction given a positive and negative tolerance level. The figure shows that the RF model, Heuristic 2 and Heuristic 1 have lots of trips, respectively 31%, 28% and 26%, where the max load is largely underestimated (more as 10 people underestimated). Beside that the errors seem to be evenly distributed around 0. Further research is needed to investigate the causes and implications.

Our passenger forecasting model performs slightly better as the model by Ohler et al. (2017). However, the relatively high RMSE (compared with the individual boarding and alighting model) and the underestimation suggest that thus method is suboptimal.

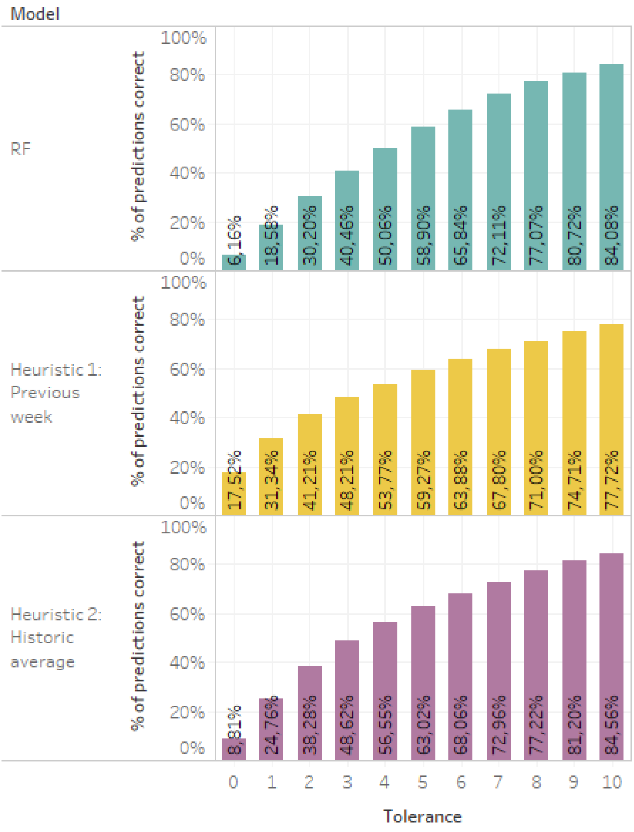


Figure 6.8: Percentage of predictions with an absolute error within the tolerance . The models are trained with data from partition 2 (all-lines/Workday 8 AM) and are tested with g554-1-0 trips in the morning peak (partition 1).

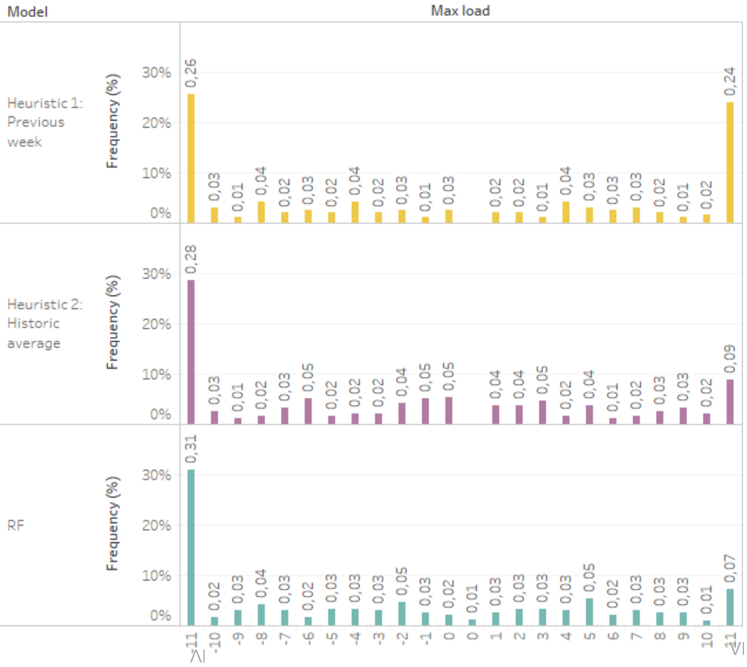


Figure 6.9: The percentage of max load predictions within a range of the true max load of the bus trip. A positive range denotes overestimation.

7 Discussion

7 Discussion

In this section we will discuss the results, procedure and the implications of the assumptions that were made.

This study has shown that forecasting the number of passengers in the average scenario can be improved in regard with the current practices. However, there is still room for improvement. It is yet unclear how well the suggested forecasting models perform in uncertain scenario's such as large events. Furthermore, when encountering such scenarios, it should be noted that the currently included features do not allow for forecasting the demand of temporary lines (lines without a historic data). This is a challenge, especially since these lines are more common around large events. The factors that have the biggest impact on the current results are the selected data, the constructed and tested features, the chosen machine learning models and the manner the passenger demand was forecasted.

It was determined that the most important performance indicator is the correct prediction of the max load. Furthermore, you should take the bus size and frequency into account when interpreting the prediction error of the number of passengers; A less frequently serviced bus stop has less capacity to cope with high fluctuations in demand. Thus, the same error in forecast at a less frequently serviced bus stop should be taken more seriously than at a more frequently serviced bus stop (Pereira et al., 2015). The same goes for the type of used vehicle; each type of vehicle has a different configuration and thus a different seat and crush capacity (Van Oort et al., 2015a). Qbuzz operates a variety of bus types in Groningen Drenthe, see Appendix Y. For this research the used buses per trips and their capacities were missing, so we could not take this into account.

When looking at the final models of this study the models were prone to underestimate the demand. Currently, Qbuzz and other public transport operators would rather have capacity. Thus, this is an undesired effect.

For a big part we used the same set up for forecasting the demand as Ohler et al. (2017). They also first separately forecast the number of people boarding and alighting. Unfortunately, Ohler et al. (2017) only reported the Mean Absolute Error (MAE) for these models, which makes comparing our models to theirs a little harder. They reported a best performance for alighting passengers of 1.54 MAE and 1.86 MAE for boarding passengers. However, they also report that their best performing model regarding the prediction of the total passenger number is correct in a little over 80% of the predictions when allowing an absolute error of 10. Our best performing model scores 84.08% and is thus better. However, it should be noted that our model is scored for trips during the morning peak, whereas the models by Ohler et al. (2017) are for all days.

The machine learning models incline towards using the smaller data partitions. This tells us that it pays to train models per demand scenario (like bad weather, peak hour, holidays, big events etc.), especially when taking into account that on average, machine learning models improve when training on more data.

One of the main challenges of the master thesis was the merging of the datasets. Especially the trip planner dataset was hard to merge on trip level. This meant that we had to introduce a large buffer at the risk of noise in the final dataset. We assumed that this noise has small consequences, moreover so because we only used a subset of the line variants thereafter. However, when this assumption is incorrect, the underlying data on which the models are trained, might change. This changes the models and their performances for better or worse. The effects of the buffer might be determined by performing a sensitivity analysis.

The trip planner dataset had some other challenges, it might for instance be oblivious to really short trips because of its underlying algorithm. Each trip possibility it neglects might have a negative influence on the predictability. Furthermore, there was a clear pattern visible in the trip planner requests, with large peaks around a request interval of a multiple of 60 minutes (indicating that most users only change the hour, see Figure 4.9). This pattern was probably caused by the human interaction with the application and thus the design of the user interface. If this design changes, it might be that different patterns emerge which should then be accounted for.

Furthermore, the reliability of the recorded departure and arrival times of the buses seem concerning; often these times were missing or incoherent. The merging of the datasets, the first step in the analysis, was based on these times. This could also be the source of mismatches and could introduce noise in the final dataset.

Finally, we determined the number of people boarding and alighting (and thus the number of people onboard) solely based on smart card data. Because there are still other payment methods around, this figure is the lower bound of the true values (even though other payment methods are less used). It could be that the overestimation in Figure 6.7 is due to missing passenger counts due to the other payment methods. One could even argue, that people who do not use a smart card, are people that do not travel by public transport that often. Another consequence could be that these people consult the trip planner multiple times, causing a peak in the number of requests and thus higher predictions. This while these extra passengers would not show up in the smart card dataset.

The choice of using trip planner data may know ethical aspects. A survey conducted in Edinburgh by Islam et al. (2017) concluded that younger people are more likely to use real-time public transit information. Also, it is more likely that these sources are consulted on longer journeys. It could therefore be that whole user groups are neglected when using this data. This is an undesired side effect, especially since the models could be used for shifting bus capacities and thus redirecting public financing. The algorithm should be augmented with extra data to compensate for people who don’t use a trip planner.

8 Conclusion

8 Conclusion

In this section we will discuss the conclusion. We will do this by answering the research questions. Finally, we will make recommendations in what directions to continue this research.

8.1 Research questions

Before we answer the main research question, we will answer the sub questions.

1. *What internal and external factors cause fluctuations in bus transport demand according to literature?*

From the literature review we found influencing factors in the following groups: Temporal, Spatial/built environment, Demand characteristics, Weather, Event, Holidays, Transit characteristics, Other mode characteristics, Socio-economic and Socio-psychological. The internal factors are the ones in the Transit characteristics group. These include for example: the line frequency, routes, travel times, fares and comfort. The other groups encompass the external factors. The most important factors are time and space.

2. *What are the opportunities and challenges of using log data from 9292 for forecasting ridership?*

9292 as trip planner data source seems promising, especially since the data could be real-time available. However, this research has shown that as of yet, the infrastructure and data structures are unfit for real-time usage; There are too many steps needed to match the datasets. These steps could be automated but it is better to optimize the underlying data structure and infrastructure since this would avoid the noise introduced by the steps and the complexity and robustness of the overall process. Moreover, using the same stop definitions and storing the trip and line planning number would make the whole process simpler and more reliable. Currently the time needed to pre-process the data is huge because of missing data fields and different (custom) definitions. For starters: it would be nice to have a session id, a *time-spent-consulting-a-travel-plan* variable and it would be helpful if the trip planner also logged the planned departure/arrival time or a delay measure. Extra features could be constructed from these, like the probability the journey advice will be followed. Furthermore, it was almost impossible to match a bus trip as suggested by 9292 with a bus trip that was operated. We had to introduce a large buffer (and thus noise) before we could match these trips.

It would be interesting to incorporate other large trip planners. However, it seemed that the design of the user interface influenced the usage by users which affects the logged data. This could make it hard to combine data from different trip planners each with its own design and typical usages.

3. *What are the opportunities and challenges of using OV-chipkaart transaction data to represent ridership?*

Using the OV-chipkaart (smart card) as stand in for the number of people boarding and alighting seemed a reasonable choice. This dataset is well defined and matches far more easily with the dataset of OV-bureau and NDOV. However, the dataset could be improved. It would for instance be nice to have the line planning number and the trip number logged. This would make the matching almost instant. Furthermore, the data structure and infrastructure at Translink could be improved in order to accommodate data analysis more easily. It would for example be nice to have the transaction data standard on trip (and chained trip) level.

4. *How does 9292 log data relate with ridership?*

The answer to this question is not unambiguous but varies with the circumstances. However, it has become clear that the 9292-log data is a bit noisy because of multiple reasons. From the tested features and their importance, we conclude that it is important to look further than the number of requests for the current bus trips; it is beneficial to aggregate the requests over a short period of time.

We will now answer the main question;

Can one forecast short-term ridership of buses using data containing the consulted travel advices from a widely used trip planner for public transport and what accuracy can one achieve in different scenarios?

We only had time to test one scenario. We chose the morning peak for the line variant g554-1-0; A bus line from Roden, via a park and ride and Groningen Central Station to Beijum. From the trip planner usage data we extracted multiple features. Several seemed important, like the historic average of number requests starting at a certain stop and the number of requests for a stop aggregated over a few hours. These features performed better than features not aggregated over time. This could be caused by the noise in the current 9292 usage data due to the absence of a session id. It is currently unknown if one user makes 100 requests for 1 trip or if there are indeed 100 unique users requesting a trip. An anonymized session id could change this.

We decided to build a separate forecasting model for the number of people boarding and a separate model for the number of people alighting. Using these two models we could make predictions for the total number of people onboard. We trained and compared 5 different machine learning models for the boarding and alighting model: multiple linear regression, decision tree, random forest, neural network and support vector regression. The trained models seemed to be performing quite good compared to two simple base rules: 1 forecast the same number of people boarding/alighting as last week and 2 similar to 1 but using the average of the previous weeks (historic average). Random forest shows the best performance for both the boarding as well as the alighting model with a RMSE of respectively 2.55 and 2.20 (R^2 of 0.76 and 0.76).

However, when combining the boarding and alighting model, the prediction error increases a lot. The best performing passenger model is heuristic 2 with a RMSE of 8.60. The random forest passenger model performs second best with a RMSE of 8.72. When looking at the percentage of correct predictions the passenger model using heuristic 2 is also superior: When looking at the percentage of trips correctly forecasted within an absolute error of 5 passengers, heuristic 2 outperforms the random forest model with 84.08% against 58.9%. When looking at the percentage of correct max loads predictions of trips – the most important indicator for adjusting the size of the bus –, the forecasts of heuristic 2 and the random forest model severely underestimated (more than 10 passengers lower as the real value) the max load for more than 27% of the trips. The fact that the model based on a simple rule outperforms more advanced machine learning models suggests that a different approach is needed. Nevertheless, the passenger model still performs better as the models reported by Ohler et al. (2017).

It should be noted that the features were not solely constructed from the trip planner dataset; Some historical averages from the smart card data and some transit characteristics from the bus data dataset were used.

8.2 Recommendations

In this section we will discuss the recommendations for science and practice.

8.2.1 Practice

The merging of the datasets should be made easier. Both 9292 and Translink should log the *trip number* and the *line planning number*. These two features, together with the *stop* and the *date of operation*, would make the merging almost instant and performing a data analysis much easier. Furthermore, it would be helpful if 9292 would use individual stops (like NDOV and OV-bureau) instead of the custom defined stop clusters. 9292 should also consider adding a session id; More often than not, users of 9292 consult different advices for one trip. The user will at most follow only one of these travel advices, the other advices are noise. If there was an anonymous session id stored with each log, you could determine more easily the probability if an advice is noise or not. Especially if you also log the time the page is shown. You could also convert this into a classification problem in the sense that you have to determine the likelihood that an individual traveler is going to conform to the suggested trip or not.

9292 should do some research towards the user groups and the typical usages of their trip planner. This would help to understand the noise patterns of the trip planner dataset better and could help to construct better features.

Translink has no common practice for aggregating transactions to trips and trips to journeys, their current infrastructure is not designed for this feature. However, for data analysis it would be key to have this information readily available and that this information is reliable. We recommend therefore that Translink adapts its infrastructure.

8.2.2 Science

Further research should focus on new methods and models to predict the number of passengers on board directly. The currently tried method of first predicting the number of people boarding and alighting was unsuccessful. You could try more combinations between the boarding and alighting models. Furthermore, other machine learning methods, like fuzzy logic (which incorporates human logic), Recurrent Neural Networks (which is able to handle time series data), deep neural networks and hybrid methods should be investigated. The current models could be improved by using grid search on a larger hyper parameter space. It could also be interesting to redefine the problem as a classifying problem by using bins for the variable to be predicted or by using the following two classes: overcrowded and overcapacity.

When continuing this project we recommend to analyze the characteristics of each bus line in the different scenarios (bad weather, public holiday, peak hour, during large events, etc.) in order to develop a method to classify a trip and assign the corresponding model to this trip. The current results show that it is beneficial to develop multiple models (or more complex models) to optimally forecast the demand in different scenarios.

The current models could be improved by incorporating more data. For instance data of other major trip planners or over a longer period of time. It would also be nice if this time period includes large events like the TT van Assen, Kings day and Kings Night.

More features should be investigated and included. Like built-environment information (kadaster from BAG+), shops, offices, schools and sport and culture buildings around a bus stop. Incorporate data of other modes like the presence of train stations as a feature to nearby bus stops. Try to use the change in weather as a feature instead of the absolute value itself and investigate if there is a correlation between the change in weather and the number of requests. Or determine the competitiveness of a travel advice to private and active modes using the Google maps API. This API can be used to reconstruct the travel advice and give a more detailed analysis of other modes. The current features, especially the ones that are extracted and aggregated, could be further analyzed by means of a sensitivity analysis (including the features that are binned).

Further research could also focus on the trip matching. You could use an unsupervised machine learning model to match the trips. It would then be a space time clustering approach with extra constraints. For example, the location should be a hard match, the check in and check out of the same transaction and trip planner consult should be clustered to the same bus trip and the bus line is given for the trip planner records. The difference with the currently tried matching method is that you optimize the matches of the whole system by clustering the trip planner and smart card data trips before and then assign a trip to these clusters instead of locally optimize the time difference. The new method would rely less on the data quality.

Finally, further research can focus on building models which can simulate the different scenario's. By first defining strategies on how to incorporate the different prediction models in day to day operations and by then simulating these scenario's, the improvements in (environmental) costs and service can be measured.

References

55
65
961
33
986
331
317
194
260
840
838
968
543
24
904
246
156
612
832
372
794
54
962
862
137
69
795
530
325
701
317
467
286
341
559
676
270

References

Amrit, C., Paauw, T., Aly, R., & Lavric, M. (2017). Identifying child abuse through text mining and machine learning. *Expert systems with applications*, *88*, 402-418.

Awad, M., & Khanna, R. (2015). Support vector regression. In *Efficient Learning Machines* (pp. 67-80). Apress, Berkeley, CA.

Bagchi, M., and P. White. The Potential of Public Transport Smart Card Data. Transport Policy, Vol. 12, No. 5, 2005, pp. 464–474.

BISON. (2014, January 9). *Actuele ritpunctualiteit en voertuiginformatie Koppelvlak 6*, version 8.1.1.1. Retrieved from www.reisinformatiegroep.nl

Brakewood, C., Macfarlane, G. S., & Watkins, K. (2015). The impact of real-time information on bus ridership in New York City. *Transportation Research Part C: Emerging Technologies*, *53*, 59-75.

Brakewood, C., & Watkins, K. (2018). A literature review of the passenger benefits of real-time transit information. *Transport Reviews*, 1-30.

Central Bureau for Statistics. (2017a, September 9). *Bevolking per viercijferige postcode op 1 januari 2017*. Retrieved from https://www.cbs.nl/nl-nl/maatwerk/2017/39/bevolking-per-viercijferige-postcode-op-1-januari-2017

Central Bureau for Statistics. (2017b, October 25). *Personenmobiliteit in Nederland; vervoerwijzen en reismotieven, regio's*. Retrieved from http://statline.cbs.nl/Statweb/publication/?DM=SLNL&PA=83500NED&D1=0,3-4&D2=0&D3=a&D4=a&D5=0&D6=l&HDR=G4,G5,T&STB=G1,G3,G2&VW=T

Central Bureau for Statistics. (2018a, February 9). *Gemiddelde bevolking; geslacht, leeftijd, burg. staat, regio, 1995-2016*. Retrieved from http://statline.cbs.nl/Statweb/publication/?DM=SLNL&PA=70233NED&D1=0&D2=0&D3=5,7,57,85,88,92,102,107,109,142,145-146,172,178,183,192,215,221,223,232,246,259,281-282,298,304,346-347,384,416,436-437,445-446,450,452,482,487,493,498,502,512,533-534,545,562,566,576-577,582,590,602,604,608,617,645,661,674-675,687,707,711,723,726-727,734,756-757,759,764&D4=l&HDR=G3,T&STB=G2,G1&VW=T

Central Bureau for Statistics. (2018b, May 31). *Regionale kerncijfers Nederland*. Retrieved from http://statline.cbs.nl/Statweb/publication/?DM=SLNL&PA=70072NED&D1=0,51-56&D2=0,5-16&D3=22&HDR=T,G2&STB=G1&VW=T

Chakrabarti, S. (2017). How can public transit get people out of their cars? An analysis of transit mode choice for commute trips in Los Angeles. *Transport Policy*, *54*, 80-89.

Choi, J., Lee, Y. J., Kim, T., & Sohn, K. (2012). An analysis of Metro ridership at the station-to-station level in Seoul. *Transportation*, *39*(3), 705-722.

de Donnea, F. X. (1972). Consumer behaviour, transport mode choice and value of time: some micro-economic models. *Regional and Urban Economics*, *1*(4), 355-382.

De Palma, A., & Rochat, D. (1999). Understanding individual travel decisions: results from a commuters survey in Geneva. *Transportation, 26*(3), 263-281.

Ding, C., Wang, D., Ma, X., & Li, H. (2016). Predicting short-term subway ridership and prioritizing its influential factors using gradient boosting decision trees. *Sustainability, 8*(11), 1100.

Doi, M., & Allen, W. B. (1986). A time series analysis of monthly ridership for an urban rail rapid transit line. *Transportation, 13*(3), 257-269.

Egeter, B. (1993). Systeemopbouw in stedelijke gebieden. *Report VK 5115.301, TU Delft*

Guo, Z., Wilson, N., & Rahbee, A. (2007). Impact of weather on transit ridership in Chicago, Illinois. *Transportation Research Record: Journal of the Transportation Research Board*, (2034), 3-10.

Hensher, D. A., & Rose, J. M. (2007). Development of commuter and non-commuter mode choice models for the assessment of new public transport infrastructure projects: a case study. *Transportation Research Part A: Policy and Practice, 41*(5), 428-443.

Islam, M. F., Fonzone, A., MacIver, A., & Dickinson, K. (2017, June). Modelling factors affecting the use of ubiquitous real-time bus passenger information. In *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference* on (pp. 827-832). IEEE.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013*). An introduction to statistical learning* (Vol. 112). New York: springer.

Jiang, X., Zhang, L., & Chen, X. M. (2014). Short-term forecasting of high-speed rail demand: A hybrid approach combining ensemble empirical mode decomposition and gray support vector machine with real-world applications in China. *Transportation Research Part C: Emerging Technologies, 44*, 110-127.

Kalkstein, A. J., Kuby, M., Gerrity, D., & Clancy, J. J. (2009). An analysis of air mass effects on rail ridership in three US cities. *Journal of transport geography, 17*(3), 198-207.

Khattak, A. J., & De Palma, A. (1997). The impact of adverse weather conditions on the propensity to change travel decisions: a survey of Brussels commuters. *Transportation Research Part A: Policy and Practice, 31*(3), 181-203.

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised leaning. *International Journal of Computer Science, 1*(2), 111-117.

Kuby, M., Barranda, A., & Upchurch, C. (2004). Factors influencing light-rail station boardings in the United States. *Transportation Research Part A: Policy and Practice, 38*(3), 223-247.

Li, L., Wang, J., Song, Z., Dong, Z., & Wu, B. (2014). Analysing the impact of weather on bus ridership using smart card data. *IET intelligent transport systems, 9*(2), 221-229.

Li, Y., Wang, X., Sun, S., Ma, X., & Lu, G. (2017). Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks. *Transportation Research Part C: Emerging Technologies, 77,* 306-328.

Ma, X., Wu, Y. J., Wang, Y., Chen, F., & Liu, J. (2013). Mining smart card data for transit riders' travel patterns. *Transportation Research Part C: Emerging Technologies, 36*, 1-12.

Mulley, C., Clifton, G. T., Balbontin, C., & Ma, L. (2017). Information for travelling: Awareness and usage of the various sources of information available to public transport users in NSW. *Transportation Research Part A: Policy and Practice, 101*, 111-132.

Ni, M., He, Q., & Gao, J. (2017). Forecasting the subway passenger flow under event occurrences with social media. *IEEE Transactions on Intelligent Transportation Systems, 18*(6), 1623-1632.

Ohler F., Krempels K. and Möbus S. (2017). Forecasting Public Transportation Capacity Utilisation Considering External Factors. In *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems* ISBN 978-989-758-242-4, pages 300-311.

Pereira, F. C., Rodrigues, F., & Ben-Akiva, M. (2015). Using data from the web to predict public transport arrivals under special events scenarios. *Journal of Intelligent Transportation Systems, 19*(3), 273-288.

Qbuzz (2018). *Soorten bussen*. Retrieved from www.qbuzz.nl/gd/reis-plannen/soortenbussen

Raschka, S. (2015). *Python machine learning*. Packt Publishing Ltd.

Redactie OV-Magazine (2014, 26 November). *9292 geeft nu ook actuele reisinfo*. Retrieved from https://www.ovmagazine.nl/2014/11/9292-geeft-nu-ook-actuele-reisinfo-1710/

Rodrigues, F., Borysov, S. S., Ribeiro, B., & Pereira, F. C. (2017). A bayesian additive model for understanding public transport usage in special events. *IEEE transactions on pattern analysis and machine intelligence, 39*(11), 2113-2126.

Scikit-learn developers (2018). *1. Supervised learning*. Retrieved from https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

Spears, S., Houston, D., & Boarnet, M. G. (2013). Illuminating the unseen in transit use: A framework for examining the effect of attitudes and perceptions on travel behavior. *Transportation Research Part A: Policy and Practice, 58*, 40-53.

Stopher, P. R. (1992). Development of a route level patronage forecasting method. *Transportation, 19*(3), 201-220.

Sun, Y., Leng, B., & Guan, W. (2015). A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system. *Neurocomputing, 166*, 109-121.

Teng, C. M. (1999, June). Correcting Noisy Data. In *ICML* (pp. 239-248).

Terpstra, F. P., Meijer, G. R., & Visser, A. (2004, August). Intelligent adaptive traffic forecasting system using data assimilation for use in traveler information systems. In *The Symposium on Professional Practice in AI a stream within the First IFIP Conference on Artificial Intelligence Applications and Innovations AIAI-2004, Toulouse France.*

Trimbach, M. (2018, 7 May). Bussen hebben nog vakantie, maar de scholieren en studenten niet. *Dagblad van het Noorden*, Retrieved from https://www.dvhn.nl/

Tsai, T. H., Lee, C. K., & Wei, C. H. (2009). Neural network based temporal feature models for short-term railway passenger demand forecasting. *Expert Systems with Applications, 36*(2), 3728-3736.

Upchurch, C., & Kuby, M. (2014). Evaluating light rail sketch planning: actual versus predicted station boardings in Phoenix. *Transportation, 41*(1), 173-192.

Van Der Spoel, S., Van Keulen, M., & Amrit, C. (2012, June). Process prediction in noisy data sets: a case study in a dutch hospital. In *International Symposium on Data-Driven Process Discovery and Analysis* (pp. 60-83). Springer, Berlin, Heidelberg.

Van Hagen, M. (2011). *Waiting experience at train stations*. Eburon Uitgeverij BV.

Van Oort, N., Drost, M., Brands, T., & Yap, M. (2015a, July). Data-driven public transport ridership prediction approach including comfort aspects. In *Conference on Advanced Systems in Public Transport, Rotterdam, The Netherlands*.

Van Oort, N., Brands, T., & de Romph, E. (2015b). Short-term prediction of ridership on public transport with smart card data. *Transportation Research Record: Journal of the Transportation Research Board*, (2535), 105-111.

Van Oort, N., Sparing, D., Brands, T., & Goverde, R. M. (2015). Data driven improvements in public transport: the Dutch example. *Public transport, 7*(3), 369-389.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS quarterly*, xiii-xxiii.

Wei, H. L., Billings, S. A., & Liu, J. (2004). Term and variable selection for non-linear system identification. *International Journal of Control, 77(1),* 86-110.

Xue, R., Sun, D.J. and Chen, S., 2015. Short-term bus passenger demand prediction based on time series model and interactive multiple model approach. *Discrete Dynamics in Nature and Society*, 2015.

Zhang, J., Shen, D., Tu, L., Zhang, F., Xu, C., Wang, Y., ... & Li, Z. (2017). A real-time passenger flow estimation and prediction method for urban bus transit systems. *IEEE Transactions on Intelligent Transportation Systems, 18*(11), 3168-3178.

Zhou, M., Wang, D., Li, Q., Yue, Y., Tu, W., & Cao, R. (2017). Impacts of weather on public transport ridership: Results from mining data from different sources. *Transportation research part C: emerging technologies, 75*, 17-29.

Appendices

882
778
169
524
246
377
342
511
563
489
80
994
803
944
923
713
106
585
78
996
243
806
918
123
309
697
84
471
739
436
471
177
761
344
505
110

Appendices

Appendix A

A.1 Project owners

This project is a graduation assignment for the master study of Industrial Engineering and Management at the University of Twente. The assignment will be performed for OV-bureau Groningen Drenthe. Other stakeholders that participate in the project are 9292 and Translink. The following section describes these organizations and their role in the project.

A.1.1 Translink

Translink is the coordinating cooperation behind the Dutch nationwide smart card (OV-chipkaart). Each year, around 2 billion transactions are made with the OV-chipkaart. These transactions are gathered by Translink. Translink sends these transactions to the public transport operator in question.

A.1.2 9292

9292 is a travel information provider within the public transport sector. Travelers can use the by 9292 provided services to plan their journey using public transport. 9292 informs passengers via the app and internet. Furthermore, travelers have the opportunity to request travel information by phone (call center). However, nowadays this last service is not used as often anymore. In addition, 9292 offers an API which some public transport operators use in their own trip planners. 2.6 million travel schemes are consulted daily, of which around 1000 are given by phone.

Within this project, 9292 and Translink provide the needed transaction and consulted travel information data. These data are gathered using SQL queries. Both 9292 and Translink have agreed to assist in collecting these data and have a hot desk (flexplek) available if needed.

A.1.3 OV-bureau Groningen Drenthe

OV-bureau Groningen Drenthe is the transit authority in the provinces Groningen and Drenthe. OV-bureau regulates the bus network on behalf of the governments of Groningen and Drenthe and the municipality of Groningen. As authority, OV-bureau tenders the right to operate the buses in these regions. Currently, Qbuzz and Arriva Touring. Furthermore, OV-bureau is responsible for the network design, the timetable design and the revenues for the public transport buses in Groningen.

OV-bureau is organized in four clusters: Development, Administration, Marketing & Communication and FIJAC (Finances, Control & Legal). FIJAC is also responsible for providing information and doing analyses. The assignment will be executed within this sub cluster (F Information J Analysis C).

Supervision and a desk to work on this project, are provided by OV-bureau Groningen Drenthe.

Appendix B Literature review methodology

This appendix discusses the used approach of the literature review. The literature search is conducted as recommended by Webster & Watson (2002). First the leading journals are researched for major contributions. These contributions are investigated and used for searching backward and forward; e.g. looking for interesting articles in the references and looking for interesting articles which reference the current article. The contribution, impact, logic and thoroughness are assessed per paper.

We utilize Scopus for the review. First the top 25 journals in the research field of transportation are retrieved using the sum of the CiteScore (impact in citation of journal), SJR (score for scientific prestige) and SNIP (citation impact) of the last three years. The resulting journals can be found in Table B.1.

Next, we constructed the search term to search in the titles, abstract and keywords for these journals. The search term is constructed using related/narrower/broader terms for each term in the original research question. By using these search terms, the search will be broad enough. The used search term is as follows:

(predict OR forecast*)*

AND

("public transport" OR transit)

AND

(ridership OR "number of passengers" OR usage OR demand OR patronage)

AND

(short-term OR day OR hour* OR trip OR real?time OR "real time")*

This resulted in 41 articles. These articles were assessed on relevance by reading the title and the abstract. At the end of these steps, 11 articles were selected for further investigation and by using the forward and backward searching process, the key articles are found. These articles all investigate the impact of certain factors on demand.

Journal
Economics of Transportation
IET Intelligent Transport Systems

Journal
International Journal of Logistics Management
International Journal of Physical Distribution and Logistics Management
International Journal of Sustainable Transportation
International Journal of Tourism Research
Journal of Transport Geography
Journal of Travel Research
Maritime Policy and Management
Mobilization
Public Transport
Sustainable Cities and Society
Tourism Management
Transportation Research, Part C: Emerging Technologies
Transport Policy
Transport Reviews
Transportation
Transportation Research Part A: Policy and Practice
Transportation Research Part D: Transport and Environment
Transportation Research Part E: Logistics and Transportation Review
Transportation Research Part F: Traffic Psychology and Behaviour
Transportation Science
Transportmetrica A: Transport Science
Transportmetrica B
Transportation Research Part B: Methodological

Table B.1: Top 25 journals in the research field of transportation

Appendix C Example raw datasets

C.1 OV-bureau – Bus data

concessieareacode	dataownercode	operationdate	linepublicnumber	lineplanningnumber
GD	QBUZZ	12/12/2016	75	d075
GD	QBUZZ	12/12/2016	75	d075
GD	QBUZZ	10/02/2017	550	b550
GD	QBUZZ	13/02/2017	20	d020
GD	QBUZZ	13/02/2017	550	b550

lijnnaam	tripnumber	vehicleregistrationnumber	userstopcode	stop_id
Stadskanaal - Emmen	1005	3309	2579	12700020
Stadskanaal - Emmen	1008	3309	2869	18770030
Grootegast - Leek	1402	1	3519	11381120
Meppel - Assen	1049	68	4084	15430140
Grootegast - Leek	1402	1	3519	11381120

timingpointname	haltetype	tijdhalte	userstopordernumber	targetarrivaltime
Stadskanaal, Busstation	EIND	TRUE	63	08:29:00
2e ExloÃ«rmond, Zuiderdiep 385	INTERMEDIATE	TRUE	19	08:43:00
Grootegast, Hoofdstraat 109	EIND	TRUE	28	07:30:00
Hoogersmilde, Hendrik Oostdraai	INTERMEDIATE	FALSE	28	18:11:00
Grootegast, Hoofdstraat 109	EIND	TRUE	28	07:30:00

targetdeparturetime	recordeddeparturetime	recordedarrivaltime	recordedpunctuality
00:00:00	00:00:00	08:28:08	-51
08:43:00	08:43:00	00:00:00	0
00:00:00	00:00:00	00:00:00	0
18:11:00	18:11:00	00:00:00	0
00:00:00	00:00:00	00:00:00	0

haspassed	hasstopped	tripcancelled	tripdispatched
FALSE	TRUE	FALSE	FALSE
TRUE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	TRUE
TRUE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	TRUE

Table C.1: Raw data extract containing the bus info as obtained by OV-bureau

C.2 Translink – Smart card

id	cki_id	cki_datetime	cki_location	cki_row	cko_id	cko_datetime	cko_location	cko_row	product	exception	tijdsblok	frequentie	totaal
1	7458349	11/02/2017 22:13	21957	157	7459598	11/02/2017 22:28	22119	158	1275	0	dal	1	7
2	3252324	14/11/2016 22:49	21723	87	3384769	14/11/2016 23:06	22401	89	1275	0	weekend	1	6

Table C.2: An example of the dataset obtained from Translink containing the trips made with a smart card. The shown data is the table in fictious for privacy reasons.

C.3 9292 – Trip planner data

action	request_datetime	departuredatetime	arrivaldatetime	question_type	from_halteclusternumber	to_halteclusternumber	via_halteclusternumber	from_coordinates	to_coordinates	via_coordinates	transfer_time_option	from_halteclusternumberlist	to_halteclusternumberlist	via_halteclusternumberlist	no_of_changes
secondaryadvice	26/01/2017 10:14	26/01/2017 10:14	26/01/2017 11:13	D	ut	gn					0	{1050001,9231029,gn,1871070}	{sgn,1000477,1873231,1015021}		0
android	16/02/2017 22:17	19/02/2017 03:56	19/02/2017 04:10	D		1700001					0	{gerp,1015021,1217070,1700001}	{1700279,1217070,9208732,hdr}		2

Table C.3: An example of 1 of the datasets provided by 9292 containing the journeys which are the chained trips. The shown data in the table is fictious for privacy reasons.

id	question_tulp_id	journeypart_sequence_no	transport_company	line_no	transport_type	start_cluster_number	end_cluster_number	travel_time
134199867	{AB14D8E2-120C-497C-8A57-9D0F3AC77A8A}	3	Qbuzz	1	Bus	1000158	1255465	13
134186011	{8B9D610C-D166-49A4-BCE4-04D5CF86C018}	2			Lopen als modaliteit	1000631	gn	3

Table C.4: An example of 1 of the datasets provided by 9292 containing the trips which make up the journeys. The shown data in the table is fictious for privacy reasons.

Appendix D Found anomalies in bus data

	Anomaly	Total	Percentage	Possible explanation
1	The first stop in a trip has a recorded arrival time.	397,694	2.33%	The board computer was already sending information.
2	The last stop in the trip has a recorded departure time.	203,630	1.19%	The board computer was still sending information on this trip.
3	Recorded that a bus has passed, but there is no departure time recorded.	4,852,097	28.38%	It could be that an alighting stop was involved, but nobody needed to alight.
4	Recorded that a bus did not pass but did stop.	401,370	2,35%	All in last stop of the sequence.
5	Recorded that the bus did stop, but no arrival time recorded (excluding the first stop of the trips)	23	0.00%	Because of logging 00:00:00 as no time recorded. This can be also a valid time.
6	Recorded that the bus did stop, but no departure time recorded (excluding the last stop of the trips)	20	0.00%	Because of logging 00:00:00 as no time recorded. This can be also a valid time.
7	Recorded that the bus did not pass or stop, but also that the trip was not cancelled or dispatched.	222,798	1.30%	In practice this is manual input or input by an algorithm. A big part has also no recorded times.
8	No data recorded on the passage (also no arrival or departure time), but the trip is also not cancelled.	186,051	1.09%	It could be that the bus needed to make a detour without passing this information on to traffic control. Or it could be that the driver made a decision of some kind.
9	Recorded departure time and recorded arrival time are the same and not null	2,655,512	15.53%	In the same second the message is sent or received. The operator first collects all the data before sending it to NDOV. De timestamp is determined later to prevent synchronization errors.
10	The recorded arrival time is later than the recorded departure time.	207,428	1.21%	The message for departure is received earlier as the message of arrival.
11	Recorded that the bus did not stop but recording an arrival time.	26,303	0.15%	Error in the data. Most are cancelled bus trips (see next).

	Anomaly	Total	Percentage	Possible explanation
12	Recorded that the bus did not stop or that the trip was cancelled but recording an arrival time.	6,758	0.04%	Error in the data.
13	Recorded that the bus did not pass but recording a departure time.	270,134	1.58%	Unknown.
14	The vehicle registration number changes on trip	6,920	0.00%	These are trips were the vehicles change during the trip.
15	A drop in punctualities between successive stops of 1000 seconds or more.	1,267	0.00%	An outlier in the data.
16	Recorded that a bus arrives or departs at a stop before departing the previous stop.	16,025	0.09%	Error in the data because of malfunctioning equipment.
17	The recordedpunctuality is maxed out (3600 seconds)	382	0.00%	Error in the data or malfunctioning equipment.
18	The recordedpunctuality cannot go lower (-3600 seconds)	445	0.00%	Error in the data or malfunctioning equipment.

Table D.1: The found anomalies in the bus data set provided by OV-bureau

Appendix E Bus lines traversing the study area

Line planning number	Public line number	Description
d031	31	Ommen - Hoogeveen
d131	131	Balkbrug - Hoogeveen
e035	35	Beilen - Steenwijk
e048	48	Havelte - Steenwijk
g039	39	Surhuisterveen - Groningen
g085	85	Oosterwolde - Groningen
g133	133	Groningen - Surhuisterveen
g139	139	Surhuisterveen - Groningen
g163	163	Holwerd - Lauwersoog - Groningen
g189	189	Drachten - Groningen
g637	637	Zoutkamp - Groningen

Table E.1: Bus lines crossing the border

Appendix F Stop matching

One of the steps of merging the datasets, is matching the stops. The stops from the datasets of Translink and OV-bureau can be matched directly because of the shared id. These are the same stops as maintained by *NDOV*. The stops from the 9292 dataset can't be matched directly, since they are identified differently and are defined on an aggregated level; 9292 does not make a distinction between the two (or multiple) directions the stop has but combines these stop in clusters instead. Figure F.1 shows an example; 9292 knows only one cluster, whereas Translink and Qbuzz differentiate 4 stops. Thus, before the stops can be matched, some extra steps are needed. In this chapter these steps will explained. We will first start with an analysis of the 9292 stops.

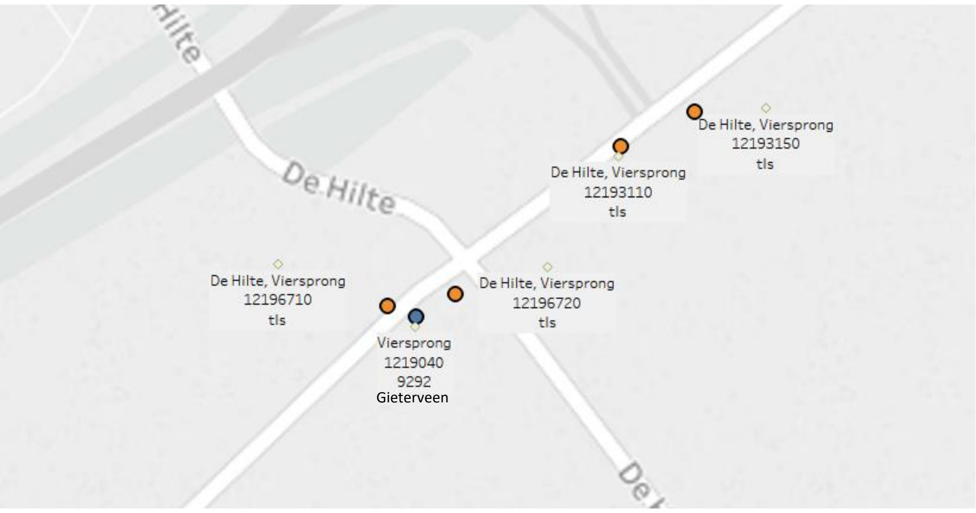


Figure F.1: Different id's and aggregation levels for top 'De Viersprong', the dataset of Translink differentiates between 4 stops, whereas the dataset of 9292 groups these in one stop cluster

F.1 9292 stop clusters

When trying to match the 9292 stop clusters and the *NDOV* stops by matching their names, around 3800 instances were not matched. This is partly because both sources use different abbreviation rules and styles, but also, because 9292 only has the second part of the name of the bus stop stored (most often this is the name of the nearby crossing street). Not the first part which is a more general identifier of the location. One good example is the bust stop “De Hilte, Viersprong” as displayed in Figure F.1; 9292 has this station stored under the name “Viersprong”. Moreover, 9292 has this stop recorded in the city “Gieterveen” and the municipality “Aa en Hunze”, and thus has no relation stored to "De Hilte". So, it is hard to match these two bus stops by name. We could try to match the second part of the names only, but unfortunately there are multiple stops called “Viersprong” in the research area. So, we would need an extra constraint, like a small distance. For this case we use an easy distance calculation: one for only in x direction and one for only in y direction.

F.2 9292 stops preprocessing

9292 has provided a list of all the stations they have stored. From this list only, the stops and stations that have the *province* parameter set to ‘Groningen’ or ‘Drenthe’ are taken into account. This results in a list of 2591 clusters. Furthermore, there is a clear difference between the cluster codes of train stations and the cluster codes of other stops, since the train stations have codes which are an abbreviation of the station name and the other stops have codes consisting of 7 numbers. Since we are not interested in train stations, these clusters are discarded. This result in a final list of 2553 bus stop clusters that are located in Groningen and Drenthe.

There are two steps needed to prepare these stops for matching: constructing a new column with the name in lower case and adjusting the RD coordinates (see A.1). The first step is to make the matching less case sensitive. Adjusting the RD coordinates is needed, because Translink has these coordinates at decimeters level instead of meters. Thus, we adjust the coordinates by multiplying with 10.

After the first matching round not all the stops where matched, because still some stops existed in the dataset which are not served by Qbuzz. These 61 stops where found by listing all the start and end clusters of the travel advices and excluding stops that have Qbuzz amongst the public transport operators. This list could contain stops which are served by Qbuzz but are not searched for by travelers during the period of the dataset.

F.3 Translink stops

Translink has provided the stops which are served by QBUZZ in Groningen and Drenthe. The biggest difference with the stop clusters used by 9292 is that the stops are defined per direction. As most stops serve bus lines in two directions there are two stops with the same name. These stops differ on the *stop_id* but have also different (RD) coordinates (see chapter A.1 for an explanation of RD coordinates) since they are most often located at different sides of the road. At hubs and some larger crossings, the transit clusters are often divided in more than 2 stops.

Two methods can be used to connect the stops used by Translink and the clusters used by 9292. First, you could go through the records of 9292 and determine per record which stop direction was used. This would result in a less aggregated dataset. However, this would require a lot of computation and introduction of noise; You would have to combine the 9292 clusters and the

Translink stops first to get the different directions and the different bus lines which serve these directional stops. Then you would have to gather all the stops which can be reached from this stop using the bus lines. Matching the destination stop of the trip with all the possible destinations per direction you can acquire the real directional stop. However, this method assumes that a stop can only be reached directly via one direction.

Method two aggregates the stops of Translink to the same aggregation level of the clusters in 9292.

We choose to implement method 2, partly because this method is less complex and partly because the stops are not needed on the smallest aggregation level for this research; We want to predict the number of passengers on a bus, so the number of passengers boarding and alighting at a stop. As long as the bus only serves one stop in a stop cluster during a trip, we have enough distinction for this analysis. Thus, as long as the rule stated before is satisfied, we can use this method. Otherwise, we will exclude the bus line.

F.4 Exploration Translink stops

In total there are 4946 stops in the dataset provided by Translink. All the stops have a unique *stop_id* and are public (column *is_public* set to *true*).

When examining the stops of Translink some things caught attention. For instance, some stops are no longer in service and can be neglected. These stops have both *embarking* and *disembarking* set to *false*. The 7 stops in question are listed in Table F.1.

stop_id	eod_stop_id	description_original	embarking	disembarking
14318410	27104	Leek, de Schutse	f	f
14318210	27102	Leek, Goldberghof	f	f
14318310	27103	Leek, Sonneborch	f	f
14318110	27101	Leek, Vredewold	f	f
10009129		Meetpunt Punc. Hoofdstation Perron U	f	f
14314210	27100	Tolbert, De Zijlen	f	f
14314220	27105	Tolbert, De Zijlen	f	f

Table F.1: Stops in the Translink dataset that are no longer in service

Furthermore, 8 stops are only used for disembarking (no stops are only used for embarking). The name (*description_*original) of some of these stops matches the corresponding stops at which embarking is allowed. For other stops, however, the suffix ‘ *uitstaphalte*’ or ‘, *uitstaphalte*’ is added to the name. 9292 aggregates the corresponding embarking and disembarking stops, so these Translink stops should be aggregated too. The 8 disembarking stops are listed in Table F.2.

stop_id	eod_stop_id	description_original	rd_x	rd_y	embarking	disembarking
15009000	20391	Assen, Station uitstaphalte	2344098	5567166	f	t
18000870	21815	Emmen, Hoenderkamp	2567731	5325470	f	t
18003000	41175	Emmen, Station uitstaphalte	2569880	5347647	f	t
10009000	22401	Groningen, Hoofdstation, uitstaphalte	2337542	5811720	f	t
10009017	22272	Groningen, Hoofdstation, uitstaphalte	2338327	5811866	f	t
10006410	24919	Groningen, v. K. Verschuurbrug	2336966	5791402	f	t
14315611	45630	Midwolde, Midwolde / A7	2213960	5780476	f	t
18040170	21621	Schoonebeek, Burg. Osselaan	2564710	5209373	f	t

Table F.2: The stops in the Translink dataset that are only used for disembarking

Also, 123 stops have a suffix with the platform description within parentheses. These suffixes are mainly used to differentiate the different platforms at hubs. Most of these stops also have the platform number recorded in the column *platform2*.

F.5 Aggregating Translink stops

We will perform the aggregation of the Translink stop using PostgreSQL. The following steps are conducted:

1. Creating a new table
2. Loading the data from the CSV
3. Removing the stops which are no longer in service by comparing it with the OV-bureau dataset
4. Creating a new column named description_without_suffix
5. Updating the new column with data from description_original taking a substring without the suffixes of the platform description and the disembarking description
6. Creating a new column named *cluster_index*
7. Updating the new column using a window function on *description_without_suffix*, where stops with the same *description_without_suffix* are grouped in one cluster with a common *cluster_index*. The RD coordinates of this newly formed clusters are the average of the stops it consists of.

The result of these steps is a table in PostgreSQL with 4939 rows and 2454 unique clusters. The frequency of the number of stops per cluster is shown inTable F.3. Most constructed clusters consist of two stops. The largest cluster is Groningen Hoofdstation which consists of 20 stops.

Count of stops	Count of clusters	Stop
1	184	
2	2166	
3	45	
4	42	
5	5	
6	8	
7	1	Groningen, Zuiderdiep
8	1	Gieten, OV Knooppunt N33/N34
12	1	Emmen, Station
20	1	Groningen, Hoofdstation

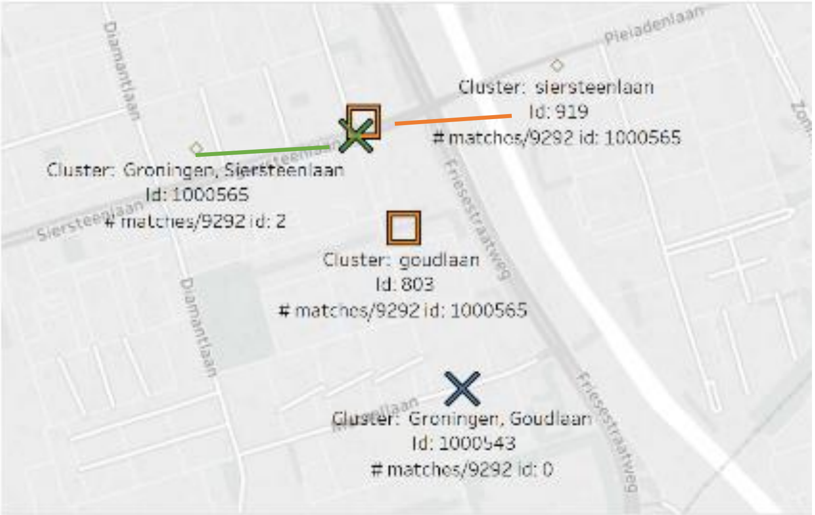
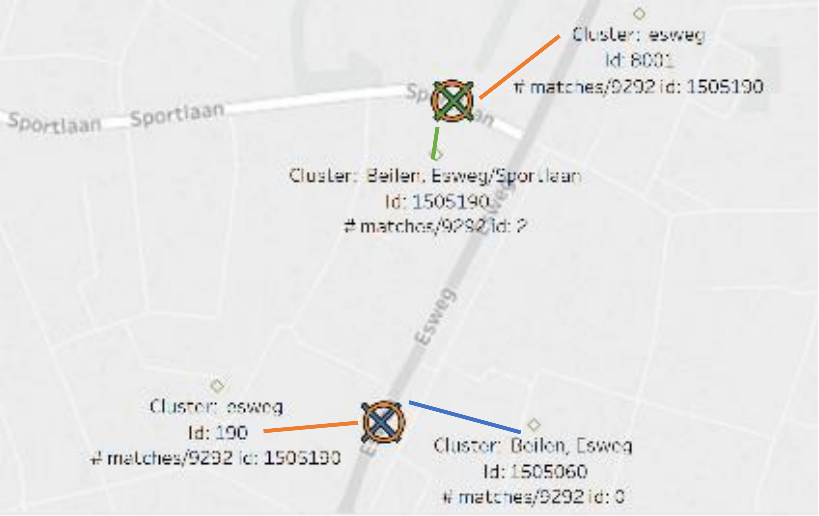
Table F.3: Clustering of Translink stops: The number of clusters with a certain number of stops. For the cluster that have a unique stop count the name is given.

F.6 Matching the stops

We matched the stop clusters of 9292 with the newly constructed stop clusters of Translink using an iterative process. The first round was matching directly by name (in lowercase without village identifier and suffixes). Because the dataset contains multiple clusters with the same name, we should choose the cluster which is closest. We used the Euclidean distance based on RD coordinates. All the stops who had a diagonal distance of less than 50 meters are accepted as a match. In total 2206 clusters are matched this way.

Stops which do not have a direct match, or where the distance exceeds 50 meters where matched based on RD coordinates (see chapter A.1 for an explanation of these coordinates); for each TLS cluster a SQL query was run selecting 9292 clusters with a x and an y coordinate which ranged 200 meters above or below. This resulted for some stops in no matches and for some in one or multiple. For each match the distance was calculated, and the stops were sorted on stop id and distance. This list was outputted to an excel sheet for validation. It was discovered that many matches, when looking at the distance, were correct. However, for some instances the distance constraint alone is not enough. For each of these instances the problem was detected and steps for correcting the mistake were prepared. 186 stops were matched this way, of which 27 clusters had to be clustered again (12 new clusters were formed) or matched differently manually.

After the matching there were still 9 unmatched stops. The unmatched stops of 9292 are listed in Table F.4Table F.4 . After further examining 5 clusters were discarded because they are (no longer) serviced by Qbuzz. The remaining 4 clusters are matched as shown in Table F.4.

Cluster information	Situation
	Legend 🟩 9292 matched 🟦 9292 unmatch.. 🟧 tls cluster 🟨 tls stop
Cluster: Groningen, Goudlaan *Code:* 1000543 *Occurrences in requests:* 23228 *Required action:* Match tls cluster goudlaan (803) to 9292 cluster 1000543 , adjust matching method to 7 and distance. Adjust # matches for 9292 cluster 1000565 and 1000543.	 Diamantlaan Pleiadenlaan Cluster: siersteenlaan Id: 919 # matches/9292 id: 1000565 Cluster: Groningen, Siersteenlaan Id: 1000565 # matches/9292 id: 2 Siersteenlaan Cluster: goudlaan Id: 803 # matches/9292 id: 1000565 Friesestraatweg Diamantlaan Cluster: Groningen, Goudlaan Id: 1000543 # matches/9292 id: 0 Friesestraatweg
Cluster: Beilen, Esweg *Code:* 1505060 *Occurrences in requests:* 2317 *Required action:* Match tls cluster esweg (190) to 9292 cluster 1505060, adjust matching method to 7 and distance. Adjust # matches for 9292 cluster 1505060 and 1505190.	 Cluster: esweg Id: 8001 # matches/9292 id: 1505190 Sportlaan Sportlaan Cluster: Beilen, Esweg/Sportlaan Id: 1505190 # matches/9292 id: 2 Esweg Cluster: esweg Id: 190 # matches/9292 id: 1505190 Cluster: Beilen, Esweg Id: 1505060 # matches/9292 id: 0

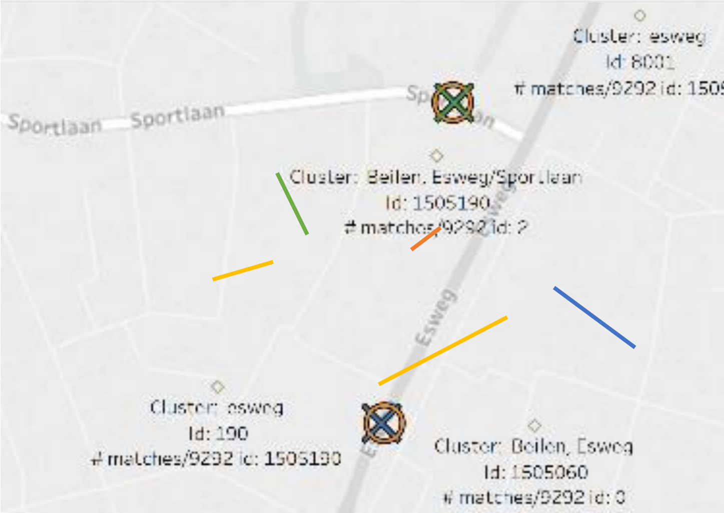
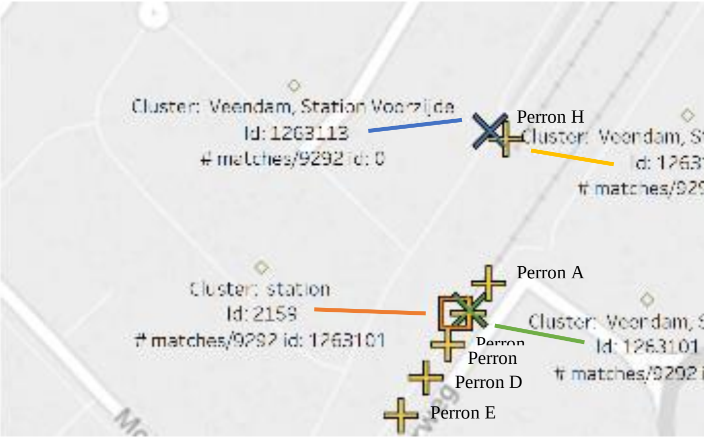
Cluster information	Situation
Cluster: Groningen, Berlageweg/Bakemastraat *Code:* 1015714 *Occurrences in requests:* 1686 *Required action:* Put tls stop 10152240 in new cluster: 8017. Adjust both old and new tls clusters. Match new cluster with 9292 cluster 1015714.	 Cluster: esweg Id: 8001 # matches/9292 id: 150 Sportlaan Sportlaan Cluster: Beilen, Esweg/Sportlaan Id: 1505190 # matches/9292 id: 2 Esweg Cluster: esweg Id: 190 # matches/9292 id: 1505190 Cluster: Beilen, Esweg Id: 1505060 # matches/9292 id: 0
Cluster: Veendam, Station Voorzijde *Code:* 1263113 *Occurrences in requests:* 1350 *Required action:* Put tls stop 12631690 in new cluster: 8018. Adjust both old and new tls clusters. Match new cluster with 9292 cluster 1263113.	 Cluster: Veendam, Station Voorzijde Id: 1263113 # matches/9292 id: 0 Perron H Cluster: Veendam, S Id: 1263 # matches/929 Perron A Cluster: station Id: 2159 # matches/9292 id: 1263101 Cluster: Veendam, S Id: 1263101 # matches/9292 Perron Perron Perron D Perron E
Cluster: Leek, De Schutse *Code:* 1431841 *Occurrences in requests:* 87	*Required action:* Belbus, no longer serviced by Qbuzz, discard cluster.
Cluster: Tolbert, De Zijlen *Code:* 1431272 *Occurrences in requests:* 55	*Required action:* Belbus, no longer serviced by Qbuzz, discard cluster.
Cluster: Leek, Vredewold *Code:* 1431811 *Occurrences in requests:* 31	*Required action:* Belbus, no longer serviced by Qbuzz, discard cluster.
Cluster: Leek, Sonneborch *Code:* 1431831 *Occurrences in requests:* 7	*Required action:* Belbus, no longer serviced by Qbuzz, discard cluster.
Cluster: Leek, Goldberghof *Code:* 1431821 *Occurrences in requests:* 2	Belbus, no longer serviced by Qbuzz

Table F.4: The remaining 9 unmatched 9292 clusters ordered by number of occurrences in request

F.6.1 Bus data stops

Next, we are going to match the stops of the dataset of OV-bureau with the stops of Translink. As stated before, the stops of these two datasets share a common id. However, there are some discrepancies. For instance, there are 57 stops missing in the OV-bureau dataset which have transaction records in the Translink dataset and request records in the 9292 datasets. Also, there are 196 stops of which the stop id was not known in the Translink dataset. Of these, 34 lie in Groningen and Drenthe as Figure F.1 shows.

These 34 stops need to be matched. Because there was no extra information available concerning these stops, some other online opensource datasets, like NDOV, GTFS info and google maps, were used to try and find the x and the y RD coordinates. Using the coordinates and information provided by these other sources 4 new TLS clusters are added. Most stops however, are added to already existing clusters. The 34 stops were added to the TLS stop dataset with an indicator that they were added later.

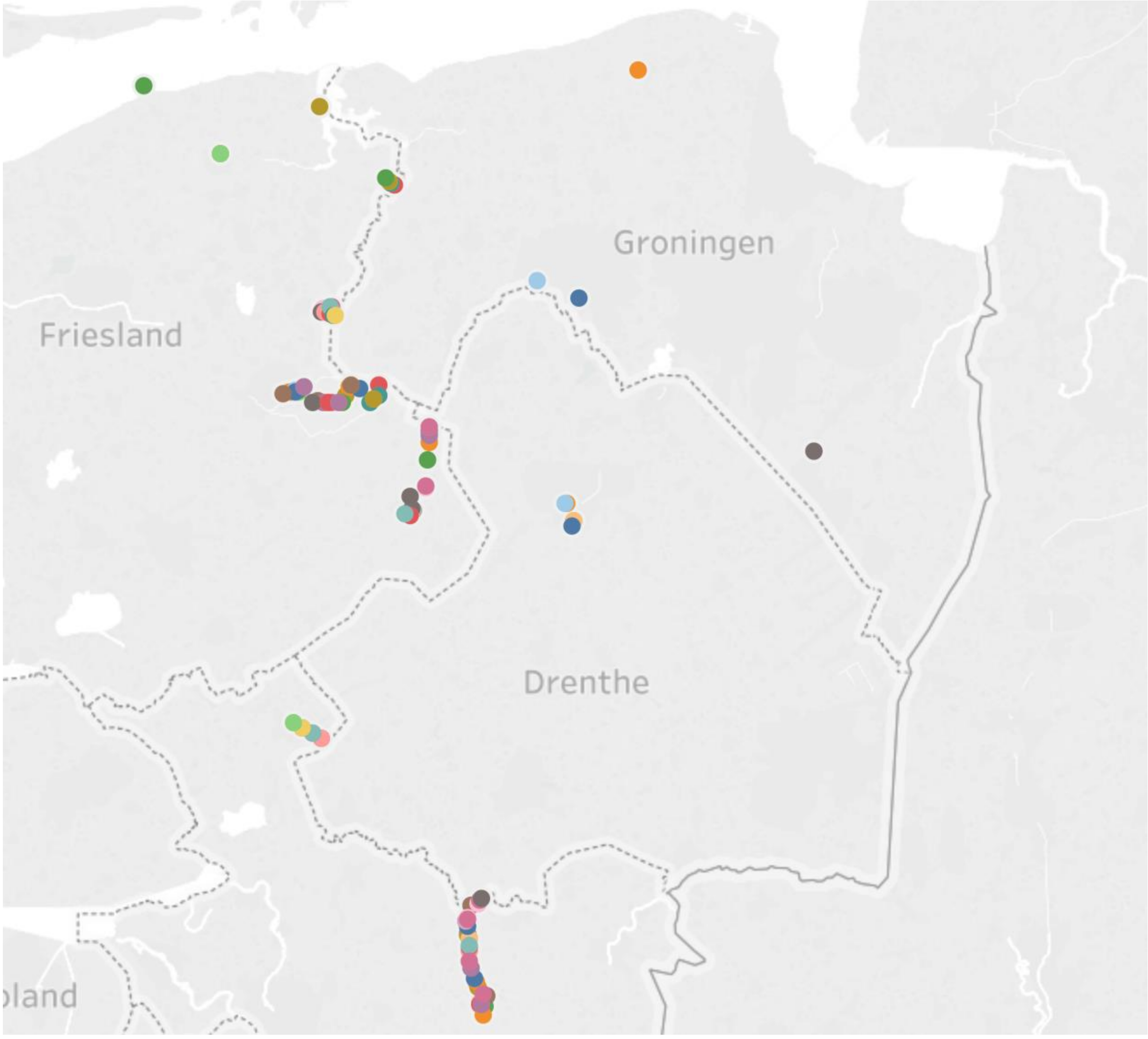


Figure F.2: The 196 stops provided by OV-bureau which are missing in the Translink dataset

F.7 Verification

This results in a merged dataset where each relevant stop in the OV-bureau and Translink dataset is grouped into clusters and where these clusters are matched with the clusters in the 9292 datasets. We will know verify if the dataset is complete.

First, we will check for the instances where the dataset has no equivalent in the other dataset. Secondly, we will test for instances were one record matches with multiple records from the other datasets. Finally, we will check for the number of occurrences of a cluster within a single bus trip. Unless the bustline makes a roundtrip, the hypothesis is that a stop can only occur once in the stop sequence.

	Number of stops/clusters
In 9292 clusters but not in constructed Translink clusters	2, one of these (cluster code of 1444150) is located just on the border of Drenthe and Friesland and therefore will be neglected. The other (cluster code of 1010035) should pair up with cluster 1340.
In constructed Translink clusters but not in 9292 clusters	2, one (1365) is on the border with Friesland and will be neglected, the stops of this cluster should be discarded too. The other (365) is on the border with Overijssel and therefore will be neglected too.
Multiple constructed Translink clusters relate to 1 9292 cluster	13, 12 clusters are duplicate because of spell errors. 1 Translink cluster, 1340, is wrongly matched as mentioned above.
Multiple 9292 clusters relate to 1 constructed Translink cluster	0
Constructed Translink clusters without a Translink stop	0
Translink stop not in cluster	0
In Translink stops but not in OV-bureau stops	124, these 124 belong to 64 clusters. Of these clusters, only 2 are served by a bus and not a belbus. These 2 clusters have other stops which are connected with the OV-bureau dataset.

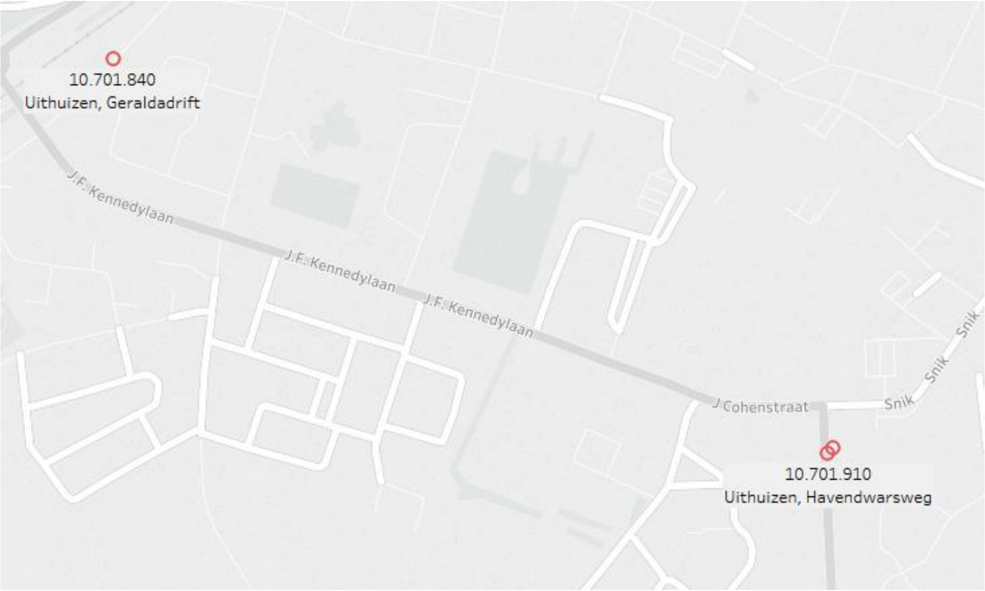
	So, via these other stops the 2 stops are connected. The 2 clusters are 974 and 2403.
In OV-bureau stops but not in Translink stops	287, 158 only when ignoring duplicates. Only 3 of these 158 stops are located within the borders of Groningen and Drenthe and are added to already existing clusters.
Stop in transaction data but not in Translink stops	0
Stop in Translink stops but not in Transaction data	0
Stop in bus data but not in OV-bureau stops	0
Stop in OV-bureau stops but not present in bus data	0
Cluster in 9292 but not in requests	4, apparently only 4 clusters did not come up in a travel request.
Clusters in request but not in 9292 clusters	76, after further investigation these clusters are present in the initial cluster dataset 9292 provided, so the missing stops are discarded during the process for one reason or another.

F.8 One cluster occurrence per trip

In total there are 64 clusters which occur twice in a bus trip. These 64 clusters are inspected regarding the original stop ids (and the distance between these stops) and the sequence order. Of the 64 clusters only two needed further inspection.

Cluster code 1070130 seems to be covering a strangely large area. Stop with id 10701840 (Uithuizen, Geraldadrift) lies far apart from the other 2 stops seems to belong to another cluster. There is a cluster, 1070080 (Geraldadrift), which lies in the immediate vacinity and contains already stops named Uithuizen, Geraldadrift.

Thus stop 10701840 is put in this cluster (constructed cluster_index = 2051).



Bus line d012 - Stadsdienst Emmen Scholen Angelslo/Meerdijk makes a consecutive stop at a stop named Emmen, Statenweg. The first stop has stop id 18000160, the second has the stop id 18007160. The stops lie on the same side of the road separated by a crossing. There are two transactions with this OD combination, but no travel requests (which is due to the fact that starting and stopping at the same cluster is useless and thus not supported).

Appendix G Variants of routes and their characteristics

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
a036	Winsum - Oldehove	a036-1-0	16	13	878	9	22	845	845
		a036-1-1	16	13	880	9	23	845	844
a042	Loppersum - Garrelsweer	a042-1-0	5	3	952	2	8	130	130
		a042-2-1	5	4	1017	2	6	130	130
a045	Loppersum - Holwierde	a045-1-0	17	14	935	8	24	744	744
		a045-1-1	17	14	928	8	25	738	738
		a045-2-0	26	26	1062	11	48	59	59
		a045-3-1	22	20	969	8	35	59	59
a245	Delfzijl - Farmsum	a245-1-0	20	10	562	0	28	1300	1299
		a245-2-1	20	10	563	0	28	1352	1352
a561	Appingedam - Oosterwijtwerd - Loppersum	a561-1-0	12	12	1145	8	22	555	510
		a561-1-1	12	12	1144	8	22	555	510
a562	Uithuizen - Loppersum	a562-1-0	19	14	901	9	27	803	803
		a562-1-1	19	14	889	9	26	803	803
a563	Ten Boer - Thesinge - Lewenborg	a563-1-0	19	16	919	6	28	546	546
		a563-1-1	19	14	888	6	28	90	90
		a563-2-1	18	16	977	6	28	456	456
a564	Appingedam - Overschild - Ten Boer	a564-1-0	15	17	1223	12	24	555	555
		a564-1-1	15	17	1227	12	24	555	555
a565	Zoutkamp - Leens	a565-1-0	18	17	1043	5	26	614	614
		a565-2-1	17	18	1128	5	25	673	673
		a565-3-1	11	14	1474	3	23	118	118
a566	Appingedam - Meedhuizen - Delfzijl	a566-1-0	23	19	871	4	28	555	555
		a566-1-1	23	19	874	4	28	555	554
a619	Appingedam - Woldendorp	a619-1-0	7	16	2815	12	26	118	118
		a619-1-1	7	16	2795	12	26	118	118
b550	Grootegast - Leek	b550-1-0	23	17	815	9	30	762	761
		b550-2-1	22	16	767	9	26	762	762
c011	Vlagtwedde - Bourtange	c011-1-0	13	6	558	5	11	708	708
		c011-1-1	13	6	540	5	11	708	707
c079	Scheemda - Zuidbroek	c079-1-0	19	13	732	7	20	195	192
		c079-1-1	19	13	727	7	20	195	192
c510	Buurtbus Veendam	c510-1-0	21	8	420	3	23	585	584
		c510-2-1	25	10	430	2	34	585	585
c512	Sellingen - Stadskanaal	c512-1-0	16	20	1366	13	26	547	547
		c512-1-1	16	20	1371	13	27	429	428

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
		c512-3-1	17	22	1395	13	30	118	118
c515	Buurtbus Hoogezand - Sappemeer	c515-1-0	22	8	420	2	22	793	793
		c515-2-1	28	11	410	2	28	793	793
c817	Finsterwolde - Bad Nieuweschans	c817-1-0	20	16	885	6	20	59	59
		c817-1-1	20	16	882	6	25	59	59
d001	Stadsdienst Emmen Emmerhout	d001-1-1	23	12	554	0	25	3010	13
		d001-2-1	20	8	472	2	20	75	
d002	Stadsdienst Emmen Angelslo	d002-1-1	24	9	426	0	28	3010	30
		d002-2-1	19	6	378	2	21	89	
d003	Stadsdienst Emmen Rietlanden	d003-1-1	23	13	614	0	31	3103	15
		d003-2-1	15	8	590	4	18	89	
d012	Stadsdienst Emmen Scholen Angelslo/Meerdijk	d012-1-0	11	4	471	3	12	576	15
		d012-2-0	15	5	406	3	16	384	
		d012-3-1	10	4	521	3	12	704	5
		d012-4-1	12	5	509	3	16	576	19
d020	Meppel - Assen	d020-1-0	56	56	1035	42	88	896	4
		d020-10-1	36	32	933	22	49	78	46
		d020-2-0	51	54	1097	41	77	982	458
		d020-3-0	24	27	1183	20	42	118	118
		d020-4-0	19	25	1395	19	35	75	1
		d020-5-0	9	5	692	0	11	65	65
		d020-6-0	24	28	1239	19	41	13	13
		d020-7-1	51	55	1112	41	77	992	234
		d020-8-1	56	57	1048	42	88	896	8
		d020-9-1	34	29	905	21	46	134	36
d021	Emmen - Assen	d021-1-0	53	43	842	31	60	1619	70
		d021-1-1	53	43	838	31	58	1619	69
d022	Zweeloo - Assen	d022-1-0	45	33	771	24	51	975	2
		d022-1-1	45	35	811	24	52	975	
		d022-2-0	27	28	1140	26	40	512	
		d022-2-1	27	31	1201	26	39	448	3
		d022-3-0	23	17	777	15	25	399	146
		d022-3-1	23	17	779	15	26	207	143
		d022-4-0	23	16	764	15	24	140	
		d022-4-1	23	18	843	15	24	204	1
		d022-5-1	35	24	719	16	36	64	
d026	Emmen - Coevorden	d026-1-0	81	49	625	17	71	1157	172
		d026-2-0	63	34	551	16	52	150	1
		d026-3-0	40	21	549	9	36	128	1
		d026-4-1	79	50	642	17	71	1069	160

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
		d026-5-1	39	21	568	9	32	101	24
		d026-6-1	41	28	711	18	38	88	13
d027	Hoogeveen - Emmen	d027-1-0	45	38	865	29	54	1476	7
		d027-1-1	45	38	869	29	54	1476	6
d031	Ommen - Hoogeveen	d031-1-0	33	15	474	11	28	1114	15
		d031-1-1	33	12	470	11	28	1202	31
		d031-2-0	19	8	493	6	16	221	221
		d031-2-1	19	8	520	6	16	273	273
		d031-3-0	26	11	451	8	22	75	2
		d031-4-0	15	6	449	4	10	13	13
d032	Meppel - Hoogeveen	d032-1-0	38	28	777	19	42	1473	364
		d032-1-1	38	28	771	19	42	1506	385
		d032-2-0	42	30	744	19	47	64	
		d032-3-1	5	1	413	0	5	64	1
		d032-4-1	3	1	771	1	3	64	
d034	Meppel - Zuidwolde	d034-1-0	16	18	1232	15	27	1440	542
		d034-1-1	16	18	1229	15	27	1394	495
d039	Koekange - Meppel	d039-1-0	14	15	1161	7	24	684	236
		d039-1-1	14	15	1164	7	23	743	298
d042	Emmen - Ter Apel - Vlagtwedde	d042-1-0	46	24	536	14	42	490	3
		d042-10-0	49	36	756	24	51	30	30
		d042-10-1	49	35	749	24	49	30	30
		d042-11-0	73	45	634	29	73	30	
		d042-12-0	30	15	541	11	29	176	
		d042-13-0	42	23	562	14	41	30	1
		d042-14-1	75	45	620	29	74	320	
		d042-15-1	44	23	538	14	41	256	
		d042-16-1	72	44	631	29	71	128	1
		d042-18-1	47	24	527	14	41	128	11
		d042-2-0	29	21	768	17	29	295	295
		d042-2-1	29	21	773	17	29	354	354
		d042-21-1	39	20	545	14	33	64	
		d042-22-1	29	14	527	10	25	64	
		d042-3-0	71	44	636	29	72	245	2
		d042-4-0	32	15	508	11	29	558	3
		d042-5-0	74	45	625	29	73	98	2
		d042-6-0	45	24	548	14	42	150	
		d042-7-0	31	15	523	11	29	382	4
		d042-8-0	70	44	645	29	72	75	
		d042-9-0	43	23	548	14	41	98	10
d044	Emmen - Schoonebeek	d044-1-0	28	18	682	14	32	1461	47

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
		d044-1-1	28	18	673	14	32	1461	47
d075	Stadskanaal - Emmen	d075-1-0	62	32	538	22	49	576	13
		d075-2-1	63	31	532	22	48	576	14
d104	Stadsdienst Emmen Bargeres	d104-1-1	24	10	436	0	28	3103	28
		d104-2-1	18	6	410	2	18	75	
d131	Balkbrug - Hoogeveen	d131-1-0	28	14	545	11	25	448	1
		d131-1-1	28	12	533	11	25	448	
d201	Stadsdienst Assen	d201-1-0	26	11	478	6	27	2741	572
		d201-2-0	19	5	320	1	13	75	
		d201-3-1	24	11	500	6	26	2741	573
d505	Stadskanaal - Gieten	d505-1-0	27	26	1030	14	40	312	312
		d505-1-1	27	27	1051	14	38	234	234
		d505-2-0	18	15	940	9	23	59	59
		d505-2-1	18	16	954	9	22	59	59
		d505-3-0	10	10	1201	6	17	59	59
		d505-3-1	10	11	1233	6	16	12	12
		d505-4-0	8	8	1248	6	13	59	59
		d505-5-1	14	13	1074	6	20	118	118
		d505-6-1	20	18	965	10	26	19	19
d626	Emmen - Klazienaveen - Schoonebeek Grens	d626-1-0	67	35	535	16	53	192	1
		d626-2-1	65	35	553	16	53	256	3
e035	Beilen - Steenwijk	e035-1-0	36	30	857	22	44	780	725
		e035-1-1	36	30	859	22	42	780	724
		e035-2-0	20	17	929	12	26	65	65
		e035-3-1	17	12	783	10	18	65	61
e036	Hoogeveen - Spier	e036-1-0	26	21	871	9	33	573	564
		e036-1-1	26	21	870	9	33	508	500
		e036-2-1	16	13	897	9	18	65	65
e037	Hoogeveen - Orvelte - Westerbork	e037-1-0	37	26	742	15	42	508	500
		e037-1-1	37	26	738	15	42	443	436
		e037-2-1	26	20	803	14	28	65	65
e048	Havelte - Steenwijk	e048-1-0	10	8	954	2	14	520	520
		e048-2-1	10	8	961	2	13	520	520
e520	Beilen - Hoogersmilde	e520-1-0	9	15	1906	10	20	590	590
		e520-1-1	9	15	1894	10	20	531	531
		e520-2-0	6	6	1398	4	10	118	118
		e520-2-1	6	6	1373	4	10	177	177
		e520-3-1	4	8	2762	7	10	59	59
e530	Hoogeveen - Drogteropslagen	e530-1-0	12	19	2175	12	30	390	390
		e530-2-0	10	8	1182	7	15	260	260

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
		e530-3-1	10	15	1747	12	25	390	390
		e530-4-1	9	9	1136	7	15	260	260
		e530-5-1	1			0	0	65	65
g012	Bad Nieuweschans - Bellingwolde - Winschoten	g012-1-0	21	14	723	9	23	1277	39
		g012-2-0	31	25	855	12	40	49	1
		g012-3-0	31	20	721	12	40	15	
		g012-4-1	17	13	813	9	21	1341	38
		g012-5-1	27	24	929	12	38	49	5
		g012-6-1	27	18	779	12	38	15	4
g013	Winschoten - Veendam	g013-1-0	42	19	475	10	38	820	289
		g013-1-1	42	19	485	10	37	1035	334
		g013-2-0	43	18	471	10	38	215	52
g014	Stadskanaal - Winschoten	g014-1-0	52	36	723	17	56	1329	19
		g014-2-0	25	15	640	13	24	256	2
		g014-2-1	25	15	644	13	26	256	1
		g014-3-1	53	36	695	17	56	1329	22
g017	Scheemda - Winschoten	g017-1-0	49	27	581	4	51	1166	107
		g017-2-0	50	31	639	4	51	1162	41
g023	Winschoten - Veendam (rijdt verder als 171)	g023-1-0	39	18	483	10	35	1782	112
		g023-1-1	39	18	494	10	33	1783	97
		g023-2-0	20	9	485	6	16	1	
g024	Winschoten - Stadskanaal - Borger - Assen	g024-1-0	102	59	592	35	101	660	13
		g024-1-1	102	59	587	35	102	660	11
		g024-2-0	101	58	596	35	101	240	9
		g024-2-1	101	58	592	35	102	240	7
		g024-3-0	51	36	721	25	57	145	
		g024-3-1	51	35	714	25	57	145	
		g024-4-0	52	23	464	17	42	104	50
		g024-4-1	52	23	462	17	42	104	49
		g024-5-0	51	22	468	17	42	30	11
		g024-5-1	51	22	467	17	42	30	10
g035	Groningen - Oldehove	g035-1-0	40	21	563	15	43	1524	11
		g035-2-0	16	9	665	7	14	234	234
		g035-2-1	16	10	671	7	17	234	234
		g035-3-1	39	21	576	15	41	1524	12
g039	Surhuisterveen - Groningen	g039-1-0	69	36	537	25	66	1575	8
		g039-2-0	51	36	731	25	67	89	
		g039-3-0	32	28	926	20	64	8	4

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
		g039-4-0	44	32	762	21	56	1	
		g039-5-1	68	36	540	25	69	1729	10
		g039-6-1	50	36	739	25	70	100	
		g039-7-1	52	28	563	20	60	60	
		g039-9-1	7	12	2154	10	33	4	4
g043	Siddeburen - Delfzijl	g043-1-0	18	15	936	9	26	562	213
		g043-1-1	18	16	943	9	25	562	214
		g043-2-0	22	16	765	10	27	270	195
		g043-2-1	22	16	766	10	25	260	260
		g043-3-0	38	31	863	9	52	128	3
		g043-4-1	37	32	891	9	51	128	11
		g043-5-1	30	23	798	10	37	70	8
g050	Assen - Groningen	g050-1-0	43	28	677	24	52	3091	21
		g050-2-0	37	24	682	21	45	192	1
		g050-2-1	37	24	673	21	45	128	2
		g050-3-0	35	22	674	20	39	1	
		g050-4-1	44	28	656	24	52	3014	23
g051	Assen - Annen - Groningen	g051-1-0	22	10	480	8	27	1767	16
		g051-1-1	22	9	472	8	25	1767	15
		g051-2-0	60	37	637	24	72	576	3
		g051-3-0	46	29	666	25	56	192	13
		g051-3-1	46	29	656	25	55	256	14
		g051-4-0	39	27	724	18	45	308	308
		g051-5-0	21	15	777	12	26	156	156
		g051-6-0	29	19	695	13	32	70	6
		g051-7-1	59	37	648	24	72	576	1
		g051-8-1	38	27	748	18	47	314	314
		g051-9-1	20	15	816	12	27	156	156
g059	Emmen - Gieten	g059-1-0	46	32	733	26	53	1411	108
		g059-2-0	30	21	736	16	33	64	
		g059-3-1	47	32	716	26	53	1411	108
g061	Groningen - Uithuizen - Delfzijl	g061-1-0	49	36	751	23	67	912	11
		g061-1-1	49	36	756	23	65	913	7
		g061-10-0	13	5	421	3	13	14	
		g061-10-1	13	4	405	3	14	14	
		g061-11-0	20	13	710	7	24	13	13
		g061-12-0	25	16	697	12	29	1	
		g061-14-1	86	69	821	27	128	64	
		g061-15-1	73	60	846	23	109	64	1
		g061-17-1	50	36	753	23	69	11	
		g061-2-0	85	69	826	27	130	320	2

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
		g061-2-1	85	69	830	27	127	320	2
		g061-3-0	32	22	728	15	44	1758	19
		g061-4-0	25	24	1022	16	37	653	652
		g061-4-1	25	24	1025	16	38	497	495
		g061-5-0	44	38	884	14	63	65	65
		g061-5-1	44	38	884	14	64	143	143
		g061-6-0	85	69	829	27	125	64	1
		g061-7-0	30	22	789	15	41	1758	20
		g061-8-0	37	33	925	18	61	64	
		g061-9-0	49	36	758	23	61	11	
g065	Zoutkamp - Groningen	g065-1-0	52	40	794	22	70	1666	15
		g065-2-0	53	37	745	22	70	612	5
		g065-3-0	32	25	816	17	42	1	
		g065-4-1	52	40	791	22	68	1610	13
		g065-5-1	53	37	741	22	68	586	4
g068	Winsum - Leens	g068-1-0	26	26	1060	10	42	1343	965
		g068-1-1	26	26	1061	10	42	1394	952
		g068-2-0	13	13	1148	8	20	91	91
		g068-2-1	13	13	1149	8	21	91	91
		g068-3-1	9	7	885	5	12	59	59
g073	Emmen - Ter Apel - Stadskanaal - Gieten	g073-1-0	96	52	561	26	93	825	14
		g073-10-0	56	27	504	20	47	29	5
		g073-11-0	36	21	612	14	34	482	7
		g073-12-0	35	19	586	14	29	1	
		g073-13-1	96	53	568	26	93	900	22
		g073-18-1	74	38	533	24	67	75	
		g073-19-1	74	37	531	24	72	14	
		g073-2-0	99	53	561	26	91	870	16
		g073-3-0	99	55	567	26	92	660	9
		g073-4-0	36	21	601	14	33	482	9
		g073-5-0	71	36	530	24	71	150	9
		g073-6-0	38	19	523	14	33	54	26
		g073-6-1	38	19	526	14	33	248	32
		g073-7-0	62	34	584	22	59	157	1
		g073-9-0	62	36	592	22	59	352	
g074	Groningen - Stadskanaal - Emmen	g074-1-0	107	73	691	51	117	1501	27
		g074-2-0	57	44	796	36	68	297	10
		g074-3-0	72	40	566	29	66	104	2
		g074-4-0	51	28	573	22	45	104	1
		g074-5-0	47	40	874	32	58	1	

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
		g074-6-1	107	71	687	51	113	1680	34
		g074-7-1	72	38	561	29	63	104	1
		g074-8-1	57	44	793	36	65	119	4
g078	Appingedam - Groningen	g078-1-0	61	31	518	20	54	1164	8
		g078-2-0	76	41	551	23	75	663	4
		g078-3-0	74	40	555	22	68	1	
		g078-4-1	61	30	513	20	56	1420	5
		g078-5-1	76	40	546	23	76	333	
		g078-6-1	77	42	554	23	85	75	
g083	Assen - Leek	g083-1-0	35	29	875	22	50	1074	313
		g083-2-1	36	29	854	22	49	1061	297
g085	Oosterwolde - Groningen	g085-1-0	22	11	543	9	17	699	211
		g085-1-1	22	11	538	9	17	904	226
		g085-2-0	35	32	971	21	48	374	2
		g085-3-0	34	32	1000	21	48	330	3
		g085-4-1	36	31	889	21	49	576	2
g086	Norg - Hoogkerk P+R - Groningen CS	g086-1-0	24	24	1157	17	42	384	4
		g086-1-1	24	24	1158	17	42	448	5
		g086-2-0	15	20	1468	15	30	64	1
		g086-3-0	10	3	533	3	11	22	
		g086-3-1	10	3	516	3	13	22	
g088	Groningen - Leek	g088-1-0	31	17	590	13	32	1026	9
		g088-2-1	31	17	593	13	32	1026	9
g089	Leek - Marum	g089-1-0	22	13	625	9	21	448	3
		g089-2-1	24	12	556	9	24	448	2
g110	Veendam - Gieten - Assen	g110-1-0	28	35	1309	24	49	1799	183
		g110-1-1	28	36	1342	24	52	1863	211
		g110-2-0	7	18	3009	12	19	1152	191
		g110-2-1	7	17	2946	12	19	1152	193
g119	Delfzijl - Winschoten	g119-1-0	25	32	1334	22	47	1625	44
		g119-1-1	25	31	1330	22	47	1649	56
		g119-2-0	1			0	0	89	1
		g119-3-1	1			0	0	89	
g133	Groningen - Surhuisterveen	g133-1-0	39	35	942	25	49	750	5
		g133-2-0	40	35	909	25	49	810	8
		g133-3-0	18	35	2107	25	49	51	
		g133-4-0	19	35	1969	25	49	55	
g139	Surhuisterveen - Groningen	g139-1-0	60	32	556	25	56	240	3
		g139-2-0	42	32	801	25	56	16	
		g139-3-1	60	32	554	25	60	60	1

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
		g139-4-1	42	32	797	25	60	4	
g163	Holwerd - Lauwersoog - Groningen	g163-1-0	28	46	1713	33	60	355	1
		g163-1-1	28	46	1711	33	58	355	2
		g163-2-0	29	43	1654	33	60	132	1
		g163-2-1	29	42	1651	33	58	132	1
g171	Veendam - Groningen	g171-1-0	44	38	890	28	65	1009	69
		g171-2-0	43	38	912	28	64	350	29
		g171-3-0	9	3	465	2	10	474	35
		g171-3-1	9	4	514	2	8	613	50
		g171-4-1	41	38	966	28	65	1412	99
g174		g174-1-0	21	10	501	6	20	1412	485
		g174-1-1	21	9	489	6	20	1412	486
		g174-2-0	69	38	571	24	75	306	34
		g174-3-0	68	38	579	24	75	100	7
		g174-4-0	16	7	515	3	18	11	
		g174-5-1	69	38	559	24	74	289	41
g178	Appingedam - Groningen	g178-1-0	41	28	718	20	47	673	3
		g178-2-0	56	38	709	23	68	320	3
		g178-3-0	41	28	721	20	48	587	10
		g178-4-1	56	39	709	23	65	384	1
g189	Drachten - Groningen	g189-1-0	22	32	1556	23	45	480	1
		g189-2-0	21	32	1633	23	45	420	7
		g189-3-0	10	5	615	4	11	484	250
		g189-3-1	10	5	609	4	10	456	358
		g189-5-1	23	30	1399	23	46	1028	3
g300	Klazienaveen - Emmen - Groningen	g300-1-0	8	58	8405	51	57	6761	48
		g300-1-1	8	57	8224	51	56	6761	50
		g300-2-0	4	27	9262	25	27	1	
		g300-3-1	42	29	711	0	52	2616	14
		g300-4-1	5	30	7514	26	27	1	
g309	Assen - Groningen	g309-1-0	15	33	2384	24	44	3271	13
		g309-1-1	15	32	2314	24	45	1699	6
		g309-2-0	9	28	3621	24	33	1705	5
		g309-2-1	9	28	3505	24	33	2763	12
		g309-3-0	14	31	2447	24	38	93	
		g309-3-1	14	31	2390	24	44	863	2
		g309-4-0	4	24	8141	21	25	576	4
		g309-4-1	4	23	7877	21	26	448	5
		g309-8-0	1			0	0	6	
		g309-9-0	1			0	0	6	

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
g312	Stadskanaal - Gieten - Groningen	g312-1-0	20	48	2538	36	54	2493	5
		g312-1-1	20	47	2489	36	55	2493	5
g402	Groningen - Vries [Nachtbus]	g402-1-1	39	39	1035	0	51	14	1
g417	Groningen - Roden - Leek [Nachtbus]	g417-1-1	25	36	1518	0	51	70	3
g418	Gieten - Groningen [Nachtbus]	g418-1-0	6	27	5520	25	26	28	1
		g418-2-1	40	31	819	25	43	28	
g419	Assen - Groningen [Nachtbus]	g419-1-0	9	27	3448	24	27	70	2
		g419-2-1	8	28	4038	24	28	70	2
g500	AirportLink	g500-1-0	11	16	1680	9	36	345	316
		g500-1-1	11	16	1660	9	39	405	316
g503	Lewenborg - Leek	g503-1-0	39	28	758	18	57	4350	18
		g503-2-0	39	29	767	19	60	2248	5
		g503-3-0	29	13	486	8	38	224	1
		g503-3-1	29	13	485	8	38	224	
		g503-4-0	17	19	1244	13	28	1	
		g503-5-1	39	29	789	18	59	4261	20
		g503-6-1	39	30	807	19	61	2259	3
		g503-7-1	27	12	473	7	32	1	
g505	Station Europapark - Annen	g505-1-0	28	30	1112	19	57	4522	95
		g505-2-0	14	11	912	3	32	2141	66
		g505-3-0	14	10	836	3	28	2028	63
		g505-4-0	26	29	1166	18	49	1	
		g505-5-1	28	29	1103	19	56	4680	104
g508	Hoogkerk - Hoogkerk	g508-1-0	47	20	450	0	55	2333	
		g508-2-0	42	16	398	2	44	1086	11
		g508-3-0	41	14	375	2	44	400	
		g508-4-0	29	10	388	2	30	1	
		g508-5-1	47	21	458	0	56	1705	
		g508-6-1	42	15	390	2	43	1152	11
		g508-7-1	46	20	470	0	56	628	
		g508-8-1	40	14	386	2	43	424	
g512	Kardinge - Driebond - Hoofdstation	g512-1-0	26	13	523	3	33	1050	7
		g512-2-1	25	13	564	3	30	1050	11
g517	Roden - P+R Hoogkerk - Groningen Zernike	g517-1-0	15	8	602	5	20	609	3
		g517-2-0	17	8	514	5	20	503	3
		g517-4-1	34	21	662	14	45	192	
g551	Zuidhorn - Zernike - Hoofdstation	g551-1-0	19	7	425	4	26	4147	20
		g551-1-1	19	7	408	4	26	4223	17
		g551-2-0	25	16	681	11	38	1800	2

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
		g551-2-1	25	16	670	11	38	1800	
		g551-3-0	16	5	399	4	27	2	1
		g551-3-1	16	6	411	4	20	4	2
		g551-5-0	8	3	471	2	10	1	
g552	Noord - Korrewegwijk - Hoornsemeer	g552-1-0	33	11	350	4	40	5873	17
		g552-2-0	29	9	346	4	29	1	
		g552-3-1	37	11	313	4	41	5888	15
g554	Beijum - Roden	g554-1-0	43	26	631	17	61	6647	65
		g554-11-1	39	25	667	17	51	1	
		g554-12-1	18	7	463	4	20	1	
		g554-2-0	26	12	513	8	38	238	
		g554-3-0	20	8	428	4	24	315	
		g554-5-0	39	25	659	17	51	1	
		g554-6-0	19	14	780	10	22	1	
		g554-7-1	42	26	646	17	61	6661	66
		g554-8-1	25	12	541	8	36	252	1
		g554-9-1	19	8	459	4	22	300	
g556	Delfzijl - Appingedam - Groningen - Haren	g556-1-0	44	40	947	28	75	2399	60
		g556-1-1	44	41	968	28	74	2469	64
		g556-2-0	31	32	1089	24	56	1126	24
		g556-2-1	31	33	1113	24	57	1120	27
		g556-3-0	43	34	830	27	61	485	11
		g556-3-1	43	34	830	27	57	569	15
		g556-4-0	44	40	963	28	75	173	
		g556-4-1	44	41	988	28	74	178	
		g556-5-0	43	34	843	27	61	35	
		g556-5-1	43	34	847	27	57	41	
		g556-6-0	14	8	619	4	16	75	2
		g556-7-0	31	32	1118	24	52	12	
		g556-7-1	31	33	1147	24	52	7	
		g556-8-0	2	0	319	0	1	1	
g557	Noord - Vinkhuizen - CS - De Wijert	g557-1-0	23	7	362	2	25	4493	23
		g557-1-1	23	7	352	2	25	4379	21
		g557-2-0	28	10	372	3	36	1408	1
		g557-3-0	11	3	355	2	10	1	
		g557-4-1	27	10	388	3	35	1344	
		g557-5-1	17	5	349	2	18	150	
g559	De Punt / De Wijert - CS - Noord / Zernike	g559-1-0	59	23	400	12	62	1874	1
		g559-2-0	65	25	392	12	61	1159	7

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
		g559-3-0	35	11	340	6	41	1963	
		g559-4-0	36	11	320	4	36	698	4
		g559-5-0	38	16	456	10	40	75	
		g559-6-1	58	23	408	12	66	1860	
		g559-7-1	64	25	399	12	63	1187	8
		g559-8-1	36	12	350	6	41	1949	
		g559-9-1	37	11	333	4	35	788	5
g564	Station Noord - Hoofdstation	g564-1-0	15	5	403	2	19	2202	
		g564-2-1	17	5	328	2	20	2127	2
g565	Zernike - Hoofdstation	g565-1-0	9	6	754	4	15	7903	42
		g565-1-1	9	6	768	4	20	7903	39
g575	P+R Haren - Gasunie	g575-1-0	4	6	2304	4	15	1125	15
		g575-1-1	4	5	1930	4	11	1125	13
g576	P+R Haren - Zernike	g576-1-0	5	12	3028	9	24	900	15
		g576-1-1	5	11	2752	9	23	975	16
g618	Winschoten - Woldendorp	g618-1-0	27	25	992	15	42	64	2
		g618-2-1	28	25	951	15	42	64	
g637	Zoutkamp - Groningen	g637-1-0	20	19	1076	12	36	147	4
		g637-2-0	42	35	876	22	69	49	
		g637-3-0	21	17	1049	12	36	45	
		g637-4-0	43	30	814	22	69	15	2
		g637-5-1	23	20	968	12	42	147	5
		g637-6-1	24	18	934	12	42	45	2
g638	Grootegast - Zuidhorn	g638-1-0	15	18	1348	9	28	49	1
		g638-2-0	16	17	1333	9	28	15	1
		g638-3-1	13	19	1661	10	30	49	1
		g638-4-1	14	18	1662	10	30	15	
g643	Winschoten - Woldendorp	g643-1-0	44	27	642	15	50	64	13
		g643-2-1	45	27	625	15	50	64	
g665	Winsum - Groningen Zernike	g665-1-0	5	15	3913	9	22	13	13
g673	Mussel - Stadskanaal	g673-1-1	7	8	1414	6	14	64	
g679	Winschoten - Groningen Zernike	g679-1-1	5	44	11043	35	58	64	
k025	Coevorden - Zweeloo	k025-1-0	23	20	922	15	29	118	118
		k025-1-1	23	20	927	15	29	236	236
		k025-2-0	22	18	885	15	24	236	236
		k025-2-1	22	18	890	15	25	118	118
k033	Hoogeveen - Coevorden	k033-1-0	43	31	742	19	49	472	464
		k033-1-1	43	23	719	19	49	472	464
k044	Harkstede-Vries	k044-1-0	35	26	770	16	42	472	472
		k044-2-1	37	24	690	16	44	413	412

Line planning number	Description	Variant of route	Total number of stops	Total travel distance [km]	mean stop distance [m]	Total Euclidean distance [km]	Mean total travel time	Total number of trips	Total trips without AVL
		k044-3-1	15	9	643	7	16	59	59
t239	Pendelbus Grootegast - Korhorn	t239-1-0	9	3	445	2	7	188	188
		t239-2-1	14	6	533	2	14	188	188
		t239-3-1	22	32	1569	20	47	8	8
		t239-4-1	5	1	336	1	3	5	5
t310	Extra: Winschoten - Hoogeveen	t310-1-0	4			58	90	32	8
		t310-1-1	4			58	89	32	8
t810	EVENEMENT DAG: Open Dag RUG	t810-1-0	11	1	319	0	34	39	10

Appendix H Rijksdriehoeksstelsel

The *Rijksdriehoekscoördinaten* (RD coordinates) is a cartesian system to denote locations in the Netherlands. All coordinates for the Netherlands are positive and the y coordinate is always bigger than the x coordinate.

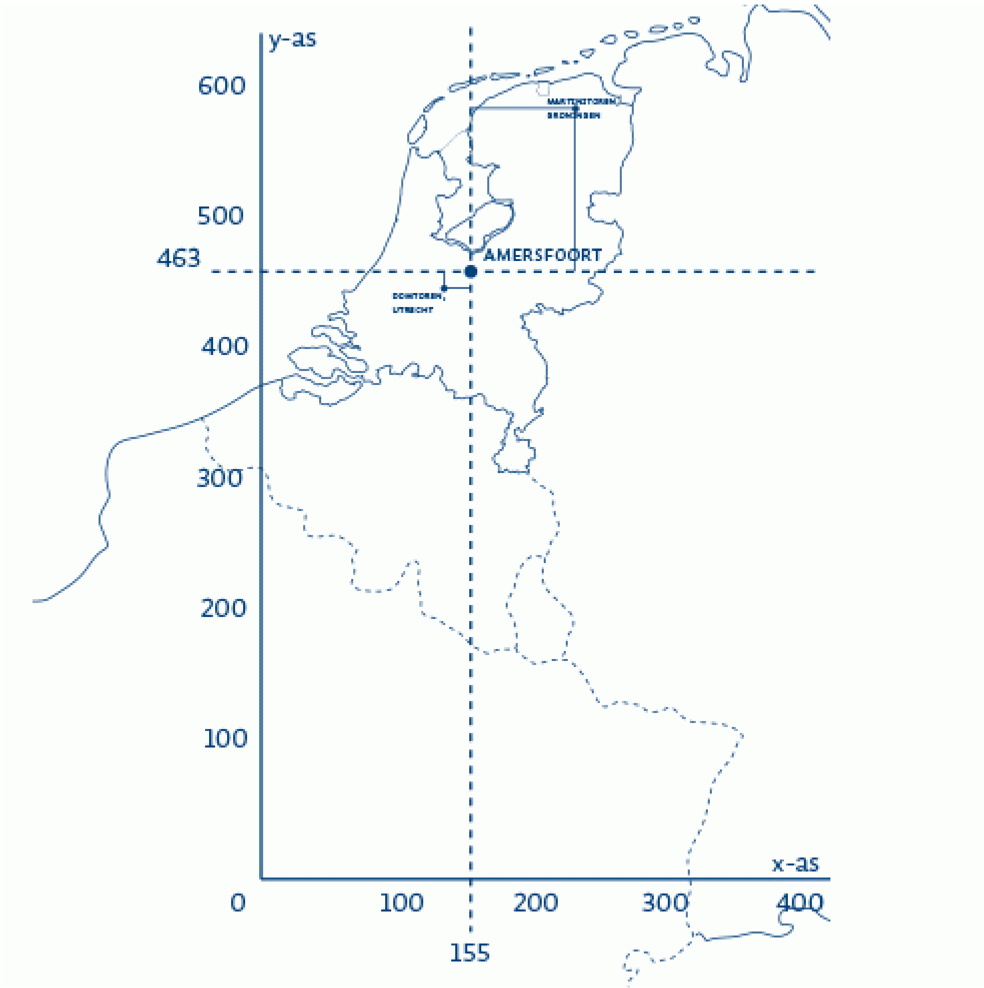
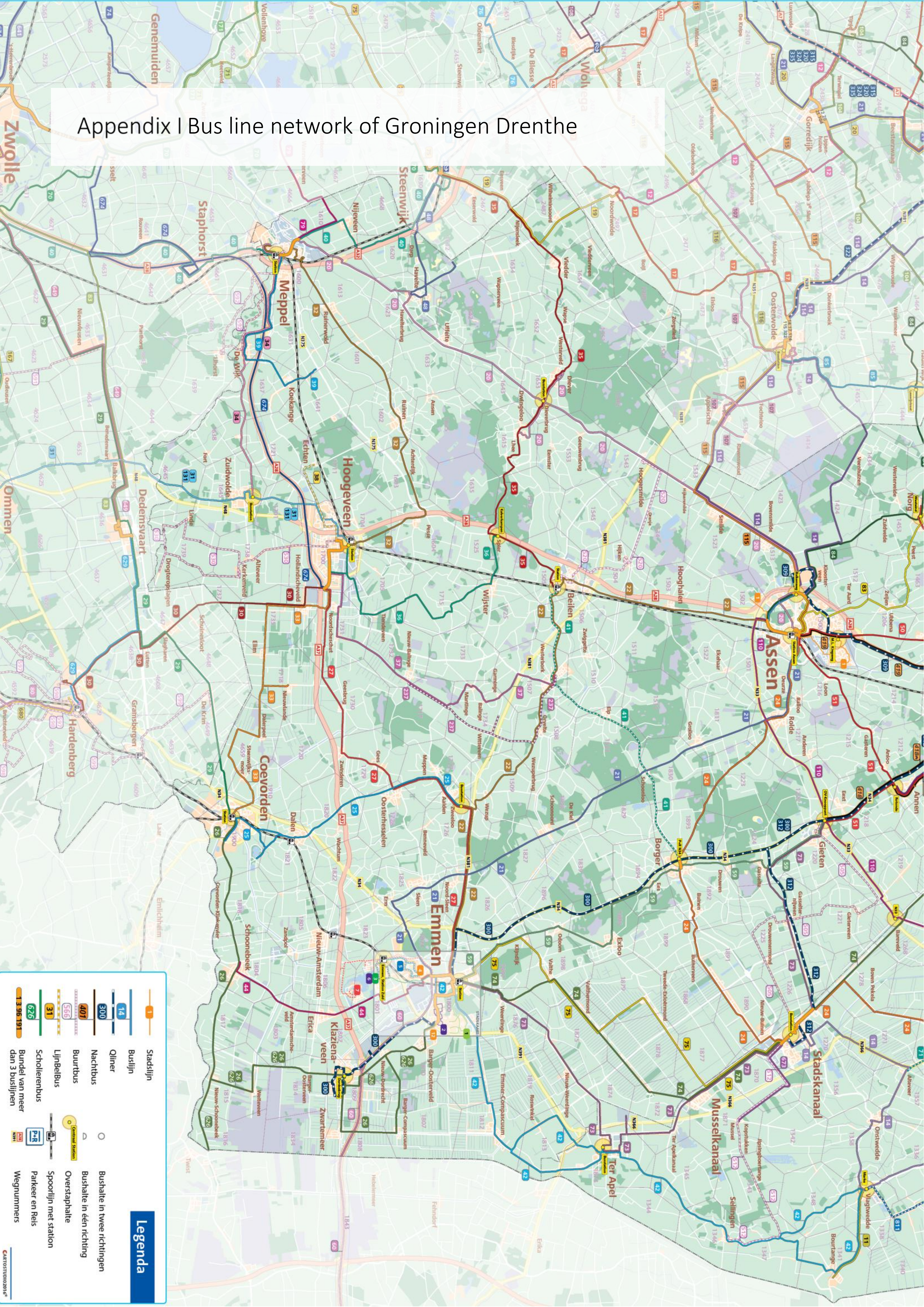


Figure H.1: The Netherlands mapped following the Rijksdriehoeksstelsel. Retrieved from https://www.kadaster.nl/rijksdriehoeksstelsel

Appendix I Bus line network of Groningen Drenthe



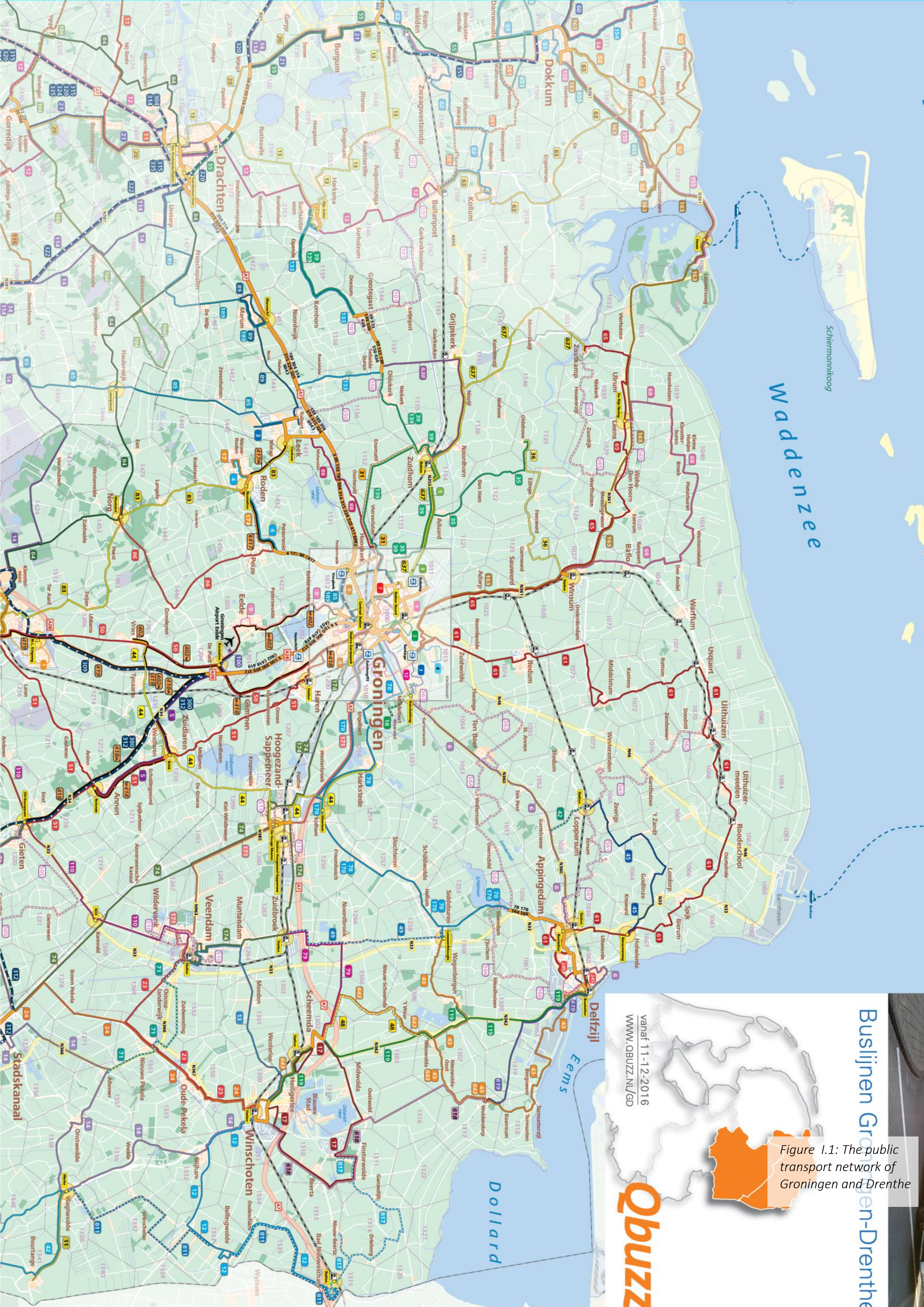


Figure I.1: The public transport network of Groningen and Drenthe

Appendix J Grouping travel advices

People are likely to use the trip planner to compare different travel options regarding departure time, route and mode. For forecasting purposes, only the chosen travel plan is valuable. The consulted travel advices which where only used for the comparison are noise and should be discarded. Unfortunately, there is no direct approach to do this: As stated before, the travel advices consulted by individuals during a session are stored separately without an identifier for the session or user.

During the exploratory data analysis, we tried to derive these separate sessions. We tried to accomplish this in two steps. The first step was determining if a travel advice was followed by another travel advice. We said this was true if a similar travel advice was consulted. A travel advice was similar if the planner, action, question_type, departure date, from-, to and via halteclusternumberlist matched. Furthermore, the advice had to be consulted at the same time or a minute later (the request datetime is rounded to minutes). We chose to let the departure date match because, we assumed that people would insert the day they are interested in and not deviate from that. At first we did not account for trips that would happen close to midnight, in which within a period of a few minutes, the date would change.

In the second step we grouped these travel advices in individual sessions.

This resulted in 11,458,259 sessions. So, 4,128,384 of the 15,586,643 consulted travel advices (about 26 %) are only part of a session and thus would not result in actual ridership. However, the largest three sessions contain over 1400 travel advices each. It is very unlikely that one individual would consult one origin destination pair that often. These sessions where over a time period of one day. Moreover, 90,363 sessions contain more than five travel advices which already a lot (11,842 sessions contain more than 10 travel advices). Finally, when looking at the request interval distribution for the trip planner requests in **Error! Reference source not found.**, we see that there is still a clear pattern of peaks around 60 minutes caused by the design and typical usage of the trip planner.

Thus, this division is not trustworthy and does not help to discard all the noise. It would be likely that only on rare occasions the number of travel advices would be higher as 5. And as many as over 1000 consults for one trip is highly unlikely. However, the 26% could be used as an upper limit: it is more likely that this number lies lower.

A side note:

- lazy people who change the question type from departure time to arrival time after they have seen the first travel advice.
- Lazy people changing the departure date after seeing the first travel advice
- Sessions around midnight
- On busy OD pairs is it impossible to differentiate users and thus sessions
- Request datetime is not trustworthy

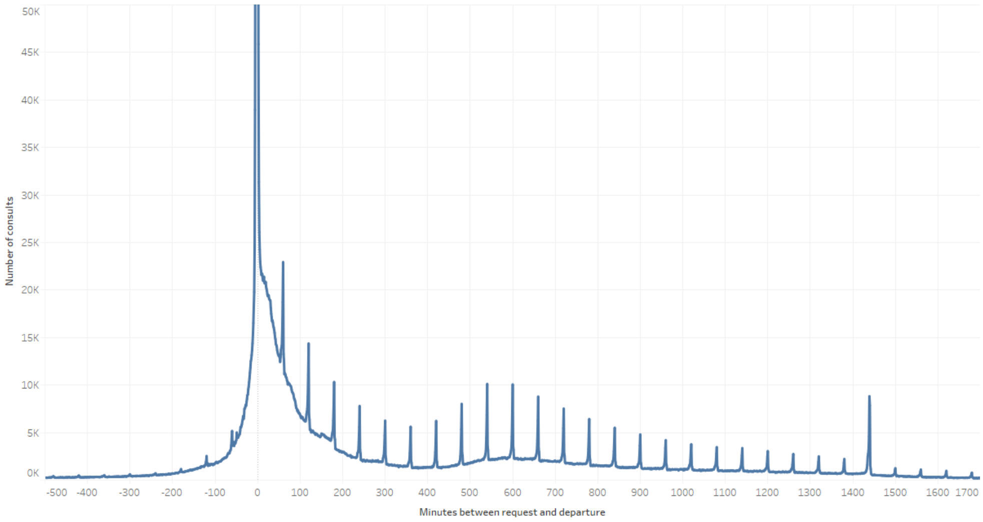
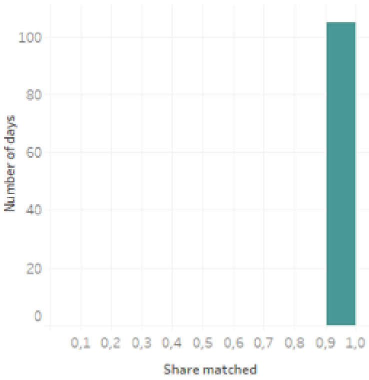
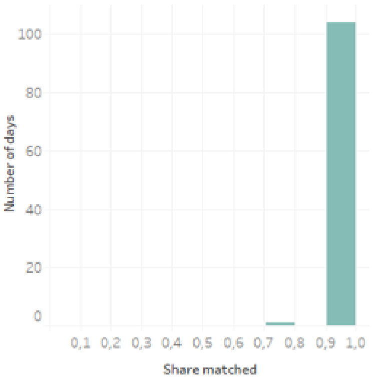
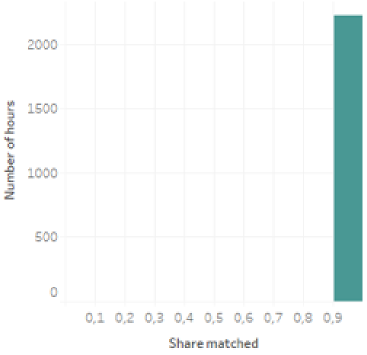
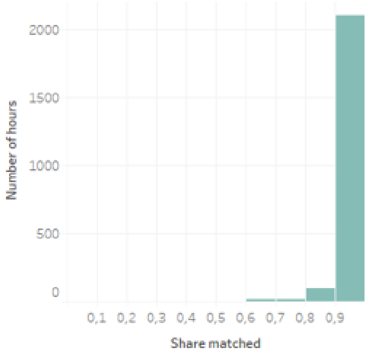
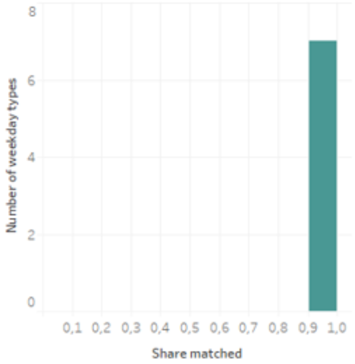
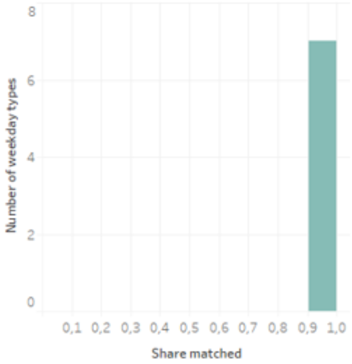
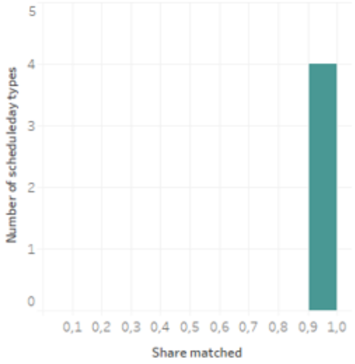
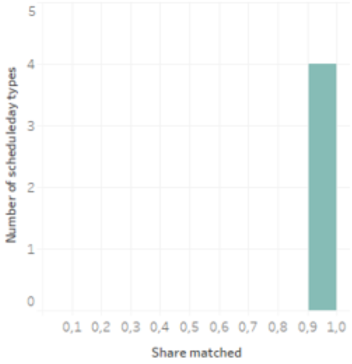
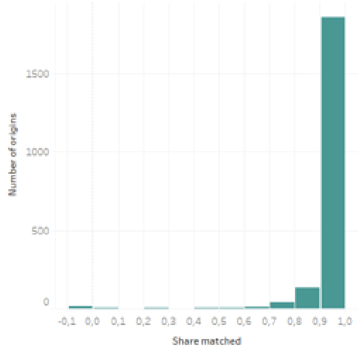
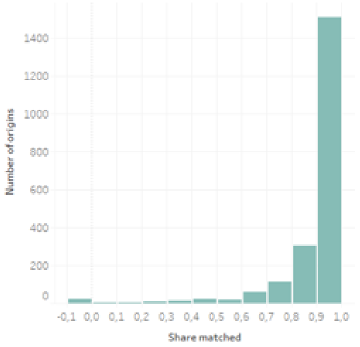
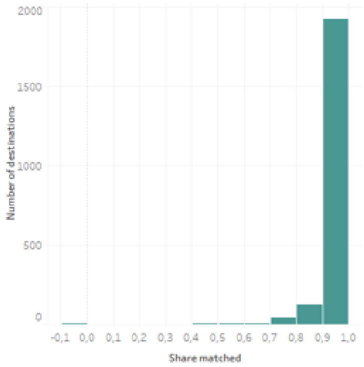
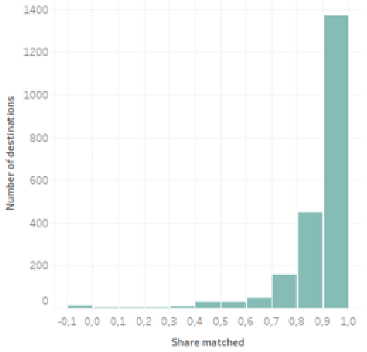
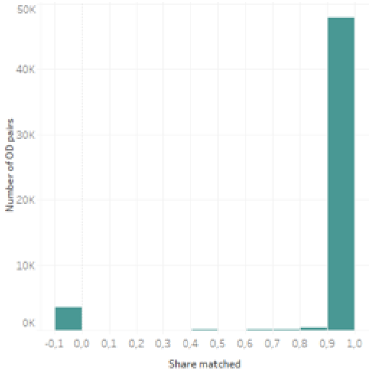
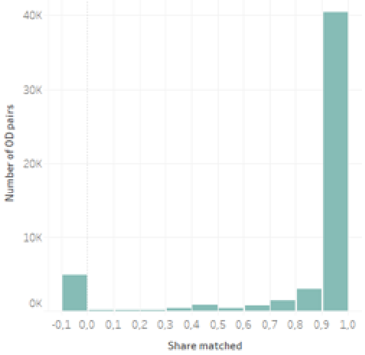
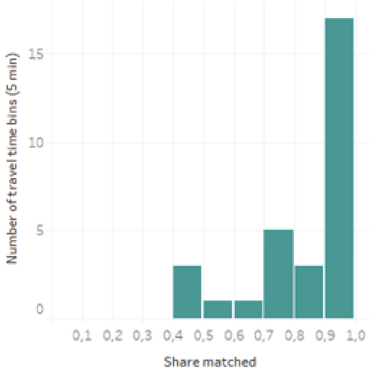
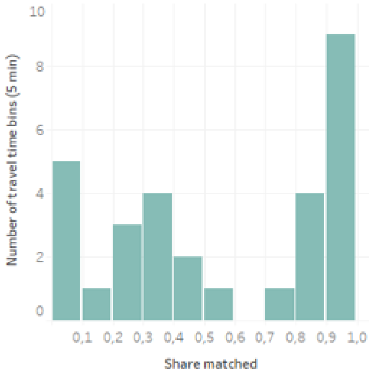


Figure J.1: The request interval distribution with the last consulted travel advice of the grouped travel advices only

Appendix K Matching smart card trips to bus trips

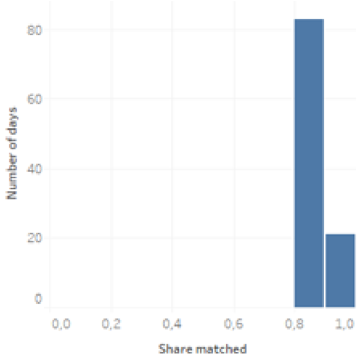
	Method 1/All	Method 2/All
Day		
Hour		
Weekday		
Schedule date		

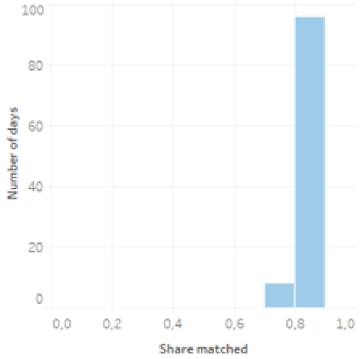
	Method 1/All	Method 2/All
Origin		
Destination		
OD pair		
Travel time		

Appendix LMatching trip planner trips to bus trips

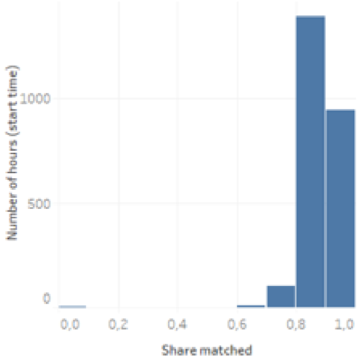
	Method 1/All	Method 2/All

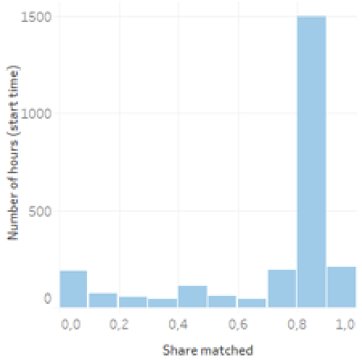
Day



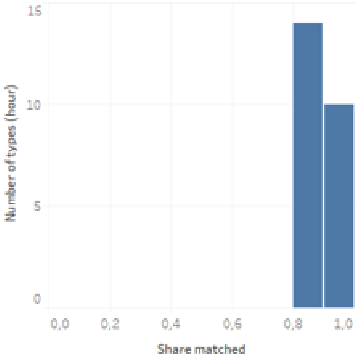


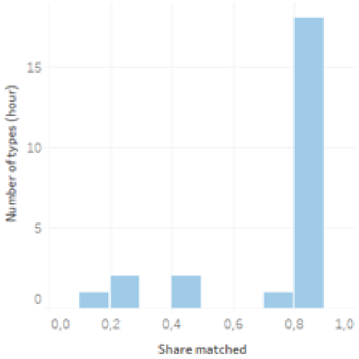
Hour



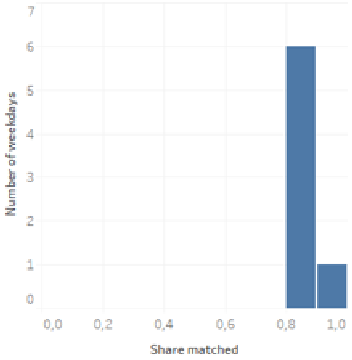


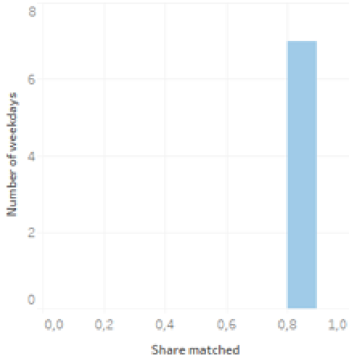
Hour type



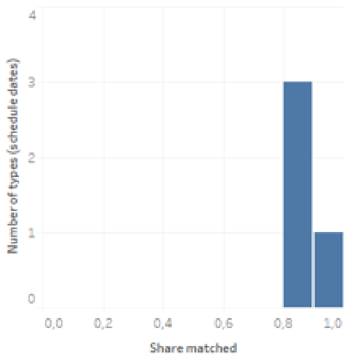


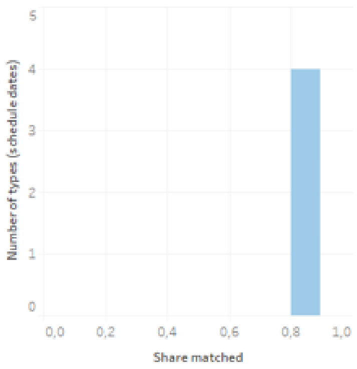
Weekday



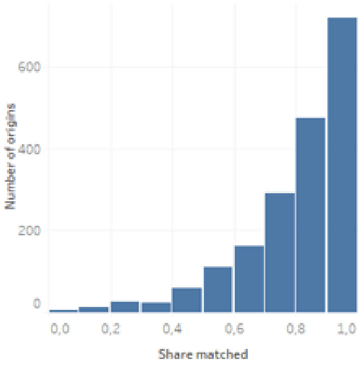


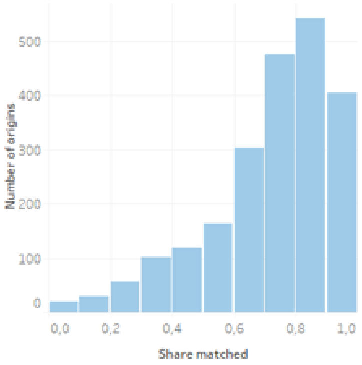
Schedule date



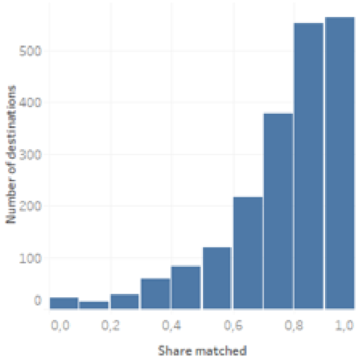


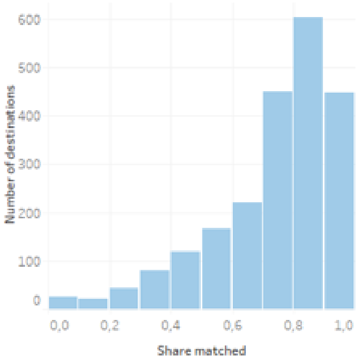
Origin



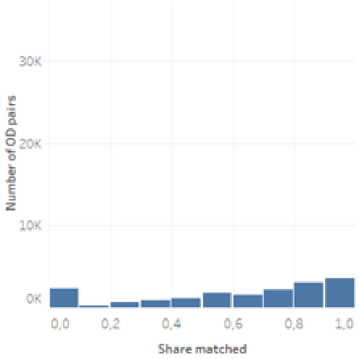


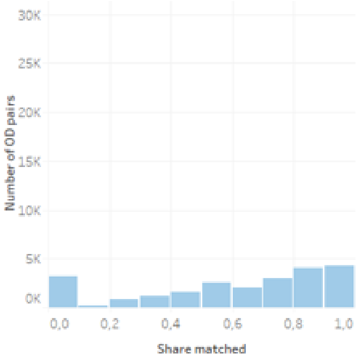
Destination



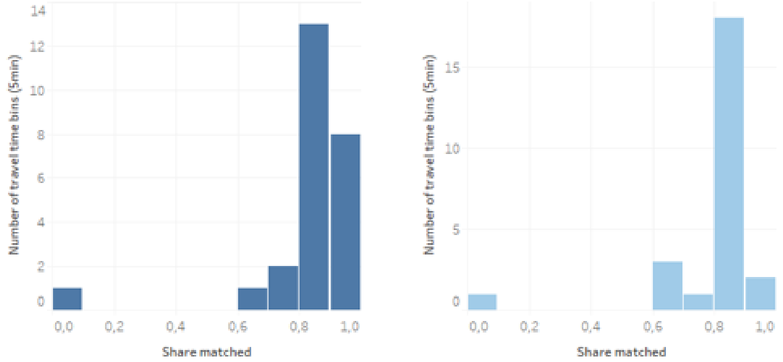


OD pair

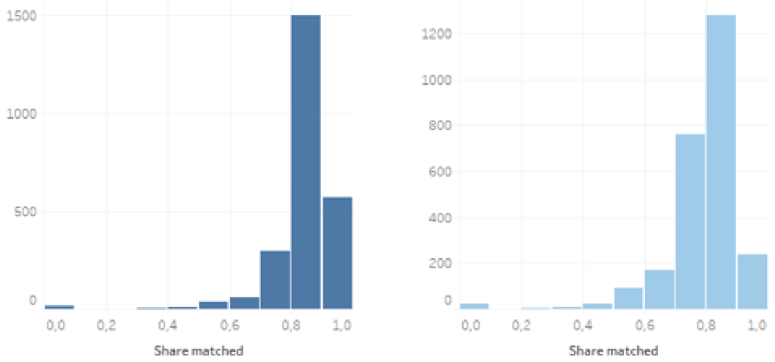




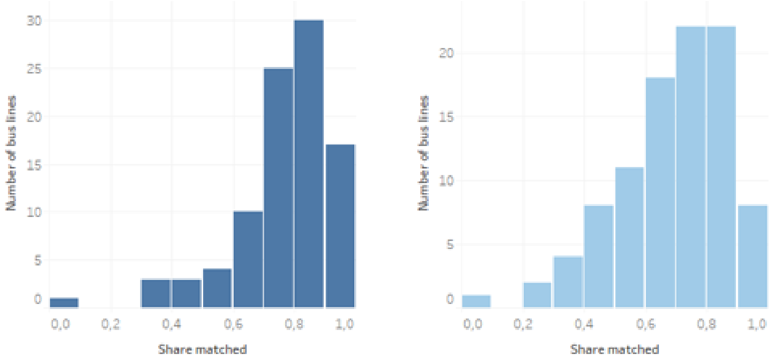
Travel time



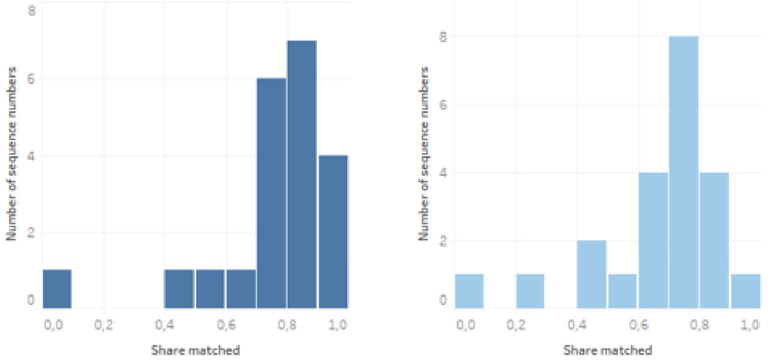
Request interval



Line_no



Journey part number



Appendix M Match performance: all vs matches on the line level

	Line planning number	Number of requests after matching using method 2	Total number of requests	Percentage matched
1	t810	1359	1481	92
2	g554	717135	832282	86
3	g551	487779	564797	86
4	g565	403949	474712	85
5	g503	689392	809977	85
6	g552	414799	492417	84
7	g505	486380	580603	84
8	g556	373704	449892	83
9	d032	43839	54380	81
10	d039	11339	14326	79
11	g557	319265	405033	79
12	d031	35337	45354	78
13	g178	50100	64737	77
14	g065	115839	153063	76
15	g078	122850	160719	76
16	g061	149395	196551	76
17	g559	532126	702543	76
18	g564	82620	108131	76
19	g300	570369	763661	75
20	g012	13519	18031	75
21	d131	8525	11318	75
22	g171	101143	136360	74
23	d012	33783	45893	74
24	g309	220628	296505	74
25	g508	239294	321963	74
26	g017	19486	26194	74
27	d001	20160	27500	73
28	g035	69243	94880	73
29	g014	43673	60208	73
30	g023	49783	67870	73
31	g088	28826	40085	72
32	g039	126191	176080	72
33	d034	24685	34123	72
34	g050	278705	392106	71
35	d022	66389	93323	71
36	d020	138101	196686	70
37	d002	45962	65287	70

	Line planning number	Number of requests after matching using method 2	Total number of requests	Percentage matched
38	g119	33729	48017	70
39	d021	58059	83465	70
40	g074	152994	221174	69
41	d044	29452	42537	69
42	d201	74283	107327	69
43	d003	28006	40393	69
44	d104	16587	24174	69
45	g051	137985	202864	68
46	d026	66098	97894	68
47	g312	130003	196039	66
48	a245	9317	14033	66
49	d626	8482	13012	65
50	g174	45558	70115	65
51	g013	14772	22820	65
52	g024	63307	99870	63
53	g139	3596	5701	63
54	g083	28573	45499	63
55	g068	25341	40124	63
56	g133	21725	34976	62
57	g500	10602	17045	62
58	g189	46831	75424	62
59	a036	4820	7966	61
60	g417	5890	9706	61
61	e530	2613	4336	60
62	g512	35615	59194	60
63	g576	2279	3767	60
64	g073	147446	247640	60
65	k033	6504	10806	60
66	c515	4023	6824	59
67	g575	942	1615	58
68	g517	25137	43216	58
69	g086	6028	10428	58
70	a562	3228	5532	58
71	d042	48106	82333	58
72	d027	52294	92354	57
73	g419	4639	8195	57
74	c510	1906	3344	57
75	a045	3000	5329	56
76	e035	8326	14776	56
77	g085	28572	53265	54
78	d075	12194	22596	54

	Line planning number	Number of requests after matching using method 2	Total number of requests	Percentage matched
79	g089	4419	8145	54
80	t310	3097	5797	53
81	g637	2442	4699	52
82	e520	3469	6927	50
83	c079	470	953	49
84	g110	131962	270678	49
85	g043	5502	11130	49
86	k025	3628	7341	49
87	e036	3258	6710	49
88	g665	20	43	47
89	e037	3885	8981	43
90	a566	1259	2948	43
91	b550	1639	3856	43
92	g059	30202	69739	43
93	a561	799	1878	43
94	a619	812	1989	41
95	g643	282	682	41
96	a565	916	2308	40
97	g163	17195	46847	37
98	g638	53	142	37
99	g618	115	314	37
100	g679	235	684	34
101	e048	233	718	32
102	g418	2790	9161	30
103	c011	772	2566	30
104	k044	4271	14664	29
105	c512	1811	6251	29
106	g673	101	422	24
107	a564	521	2246	23
108	a563	283	1403	20
109	d505	1114	5882	19
110	g402	382	2196	17
111	a042	27	197	14

Appendix N Constructed features

In the following table the features are listed with some descriptive statistics for the partition of line *g554* variant *g554-1-0* with at least 5 historic trips for trips on Monday till Friday between 2017-01-01 and 2017-03-31 which had already 5 historic trips. This partition contains 97,825 bus passages. The blue color denotes features extracted and engineered from the bus data dataset, the features with the green color are constructed from the weather dataset, the pink color denotes features extracted from the smart card dataset and the black color denotes features extracted from the trip planner dataset. Most of the features extracted from the trip planner dataset are extracted from the journey parts, however the features with a striped black color are extracted from total journey level.

The features from 28 till 66 are solely describing the process of passengers boarding the bus. These features have a counterpart (67 till 105) which solely describe the process of alighting. Furthermore features 34 till 59 are multiple subdivisions of feature 33, *start_15_total*, as are features 73 to 98 to feature 72, *end_15_total*. These subdivisions are engineered to examine the effect of journey part- and total journey characteristics on the number of people boarding or alighting. For instance, literature shows that certain aspects of the trip are important to a traveler such as speed, number of transfers, walking time and speed. Also, we are curious if people for a longer journey or who plan the journey way ahead in time are more inclined to undertake that journey. Furthermore, we want to investigate the effect of placement of the journey part in the journey: a journey part at the end of a journey with multiple transfers, is dependent on these other transit parts to function on time. And finally, from the exploratory data analysis we noted the peaks of requests at request intervals of 60 minutes, therefore we want to examine if a request with a request interval of 60 minutes is indeed noise introduced by the usage of the trip planner. Thus, we create 8 different subdivisions of the *start_15_total* and the *end_15_total* feature. These subdivisions should not be used together with the feature they are based on because of (multi)collinearity.

	Feature	Description	count	unique	mean	std	min	0.25	0.5	0.75	max
1	id	A unique identifier for the passage.	97825	97825							
2	no_historic_trips	The number of historic trips available for this bus passage (same weekday, type of schedule, tripnumber, stop and lineplanningnumber with an earlier operationdate).	97825		7.462418	1.693375	5	6	7	9	10
3	operationdate	The date of the operation.	97825	30							
4	variant_id	An identifier for the variant of the route. Unique for line planning number, direction, visited stops and stop order. Formatted as line planning number-variant-direction	97825	1							
5	Variant no	The variant of the route.	97825	1							
6	direction	The direction of the route.	97825	1							
7	lineplanningnumber	A unique identifier for the line.	97825	1							
8	clustercode_9292	An identifier for the stop cluster.	97825	43							
9	hour	The hour: Hour 0 denotes the time between 00 and 01.	97825	21							
10	scheduleday	The type of day.	97825	1							
11	weekday	The weekday ranging from Monday to Sunday.	97825	5							
12	recordedpunctuality	The recorded delay.	97825		36.36597	124.9097	-1338	-16	35	96	1131
13	stopsleft	The number of stops left to visit in the trip after this stop.	97825		21	12.40974	0	10	21	32	42
14	distanceleft	The distance (meters) left to travel before reaching the end of the line.	97825		12097.09	8889.928	0	4083	10027	22086	26486
15	tijdsblok	The time period.	97825	3							
16	prev_headway_bin	The time between the previous trip of the same line in the same direction and this trip.	97825	4							
17	next_headway_bin	The time between the next trip of the same line in the same direction and this trip.	97825	4							

		Feature	Description	count	unique	mean	std	min	0.25	0.5	0.75	max
🟦	18	before_buses_departure	The number of other buses arriving before the planned departure time (from 15 minute before up to planned departure time).	97825		5.894526	6.234034	0	3	4	7	39
	19	after_buses_departure	The number of other buses departing before the planned departure time (from 15 minute before up to planned departure time).	97825		4.637332	6.077151	0	2	2	6	40
	20	before_buses_arrival	The number of other buses arriving after the planned departure time (from 1 minute after up to 15 minutes after).	97825		5.86161	6.235815	0	3	4	7	40
	21	after_buses_arrival	The number of other buses departing after the planned departure time (from 1 minute after up to 15 minutes after).	97825		4.610263	5.957119	0	2	2	6	40
🟩	22	rainduration	The duration of rainfall per 6 minutes at stop i.	97825		0.841635	2.006539	0	0	0	0.2	10
	23	rainfall	The amount of rainfall in tenths of a mm at stop i.	97825		0.785566	2.575554	0	0	0	0.2	31.1
	24	prevrainduration	The duration of rainfall per 6 minutes at stop I for the previous hour.	97825		0.807531	1.971928	0	0	0	0.2	10
	25	prevrainfall	The amount of rainfall in tenths of a mm at stop I for the previous hour.	97825		0.757262	2.524512	0	0	0	0.2	31.1
🟪	26	passenger_delta	The number of check ins minus the number of check outs.	97825		0	2.972801	-51	0	0	0	57
	27	passenger_no	The number of passengers on the bus after visiting the stop.	97825		10.12498	10.47837	0	2	7	15	107
	28	cki_no	The number of registered check ins.	97825		1.006154	3.104349	0	0	0	1	77
	29	cki_no_historic_avg	The mean number of check ins based on the historic trips.	97825		0.977838	2.611842	0	0	0	1	49
⬛	30	start_15_total_historic_avg	The mean number of requests starting at this stop and having a request interval larger than 14 minutes based on historic trips.	97825		0.679744	1.998501	0	0	0	1	34
	31	historic_residual_start_15_total	The difference between the historic mean and the current total requests starting at this stop	97825		-0.07451	1.235241	-18	0	0	0	26

		Feature	Description	count	unique	mean	std	min	0.25	0.5	0.75	max
■			(start_15_total_historic_avg minus start_15_total).									
■	32	requests_per_cki_historic_avg	The historic ratio between requests and check ins.	97825		86.11536	137.0047	0	0.888889	1	143.571	1000
■	33	start_15_total	The number of requests starting at stop i and having a request interval larger than 14 minutes.	97825		0.605234	1.958665	0	0	0	0	43
■	34	start_15	The number of requests starting at stop i and having a request interval between 15 and 29 minutes.	97825		0.072568	0.379947	0	0	0	0	10
■	35	start_30	The number of requests starting at stop i and having a request interval between 30 and 59 minutes.	97825		0.09424	0.467371	0	0	0	0	12
■	36	start_60	The number of requests starting at stop i and having a request interval between 60 and 119 minutes.	97825		0.087708	0.432274	0	0	0	0	19
■	37	start_120	The number of requests starting at stop i and having a request interval between 120 and 359 minutes.	97825		0.088249	0.456874	0	0	0	0	17
■	38	start_360	The number of requests starting at stop i and having a request interval between 360 and 1440 minutes.	97825		0.192538	0.825069	0	0	0	0	28
■	39	start_1440	The number of requests starting at stop i and having a request interval between 1440 and 2879 minutes.	97825		0.026578	0.223691	0	0	0	0	22
■	40	start_2880	The number of requests starting at stop i and having a request interval larger as 2879 minutes.	97825		0.043353	0.333484	0	0	0	0	20
▨	41	start_15_j_walking_0	The number of requests starting at stop i with a walking time smaller than 20 minutes in the total travel advice.	97825		0.585883	1.911624	0	0	0	0	42
▨	42	start_15_j_walking_20	The number of requests starting at stop i with a walking time larger than 19 minutes in the total travel advice.	97825		0.019351	0.191098	0	0	0	0	10
▨	43	start_15_j_firstlegqbuzznot	The number of requests starting at stop i and where the request is not the first transit part in the total travel advice.	97825		0.206174	1.308872	0	0	0	0	36

	Feature	Description	count	unique	mean	std	min	0.25	0.5	0.75	max
44	start_15_j_firstlegqbuzz	The number of requests starting at stop i and where the request is the first transit part in the total travel advice.	97825		0.39906	1.121592	0	0	0	0	33
45	start_15_j_multiple_60not	The number of requests starting at stop i for which the request interval of the total travel advice is not a multiple of 60 minutes.	97825		0.529486	1.66739	0	0	0	0	40
46	start_15_j_multiple_60	The number of requests starting at stop i for which the request interval of the total travel advice is a multiple of 60 minutes.	97825		0.075748	0.467157	0	0	0	0	16
47	start_15_j_transittransfer0	The number of requests starting at stop i which have 1 transit part in the total travel advice.	97825		0.255937	0.853859	0	0	0	0	33
48	start_15_j_transittransfer1	The number of requests starting at stop i which have 2 transit parts in the total travel advice.	97825		0.240337	1.01943	0	0	0	0	26
49	start_15_j_transittransfer2	The number of requests starting at stop i which have 3 transit parts in the total travel advice.	97825		0.075257	0.493621	0	0	0	0	20
50	start_15_j_transittransfer3	The number of requests starting at stop i which have 4 transit parts in the total travel advice.	97825		0.025178	0.25327	0	0	0	0	15
51	start_15_j_transittransfer3morethan	The number of requests starting at stop i which have more than 4 transit parts in the total travel advice.	97825		0.008525	0.140202	0	0	0	0	13
52	start_15_j_speed50smallerthan_ distance100morethan_not	The number of requests starting at stop i for which the total travel advice has not a Euclidian speed smaller than 50 and a Euclidian distance equal to or more than 100 km.	97825		0.597996	1.943286	0	0	0	0	43
53	start_15_j_speed50smallerthan_ distance100morethan	The number of requests starting at stop i for which the total travel advice has a Euclidian speed smaller than 50 and a Euclidian distance equal to or more than 100 km.	97825		0.007237	0.130143	0	0	0	0	15

		Feature	Description	count	unique	mean	std	min	0.25	0.5	0.75	max
▨	54	start_15_j_speed0biggerthan	The number of requests starting at stop i for which the speed of the total travel advice is smaller than 10.	97825		0.077393	0.422414	0	0	0	0	14
▨	55	start_15_j_speed10biggerthan	The number of requests starting at stop i for which the speed of the total travel advice is bigger than 9.	97825		0.527841	1.772619	0	0	0	0	40
■	56	start_15_p_traveltime0biggerthan	The number of requests starting at stop i and having a travel time between 0 and 9 minutes.	97825		0.169088	0.887053	0	0	0	0	31
■	57	start_15_p_traveltime10biggerthan	The number of requests starting at stop i and having a travel time between 10 and 19 minutes.	97825		0.235032	1.081666	0	0	0	0	32
■	58	start_15_p_traveltime20biggerthan	The number of requests starting at stop i and having a travel time between 20 and 29 minutes.	97825		0.133902	0.572416	0	0	0	0	22
■	59	start_15_p_traveltime30biggerthan	The number of requests starting at stop i and having a travel time larger as 29 minutes.	97825		0.067212	0.419573	0	0	0	0	23
■	60	prev_trip_start_15_total	The number of requests from the previous trip starting at stop i and having a travel time larger as 15 minutes.	97825		0.596177	1.936965	0	0	0	0	43
■	61	next_trip_start_15_total	The number of requests from the next trip starting at stop i and having a travel time larger as 15 minutes.	97825		0.601922	1.956459	0	0	0	0	43
■	62	prev_stop_start_15_total	The number of requests starting at stop i-1 and having a travel time larger as 15 minutes.	97825		0.605234	1.958665	0	0	0	0	43
■	63	next_stop_start_15_total	The number of requests starting at stop i+1 and having a travel time larger as 15 minutes.	97825		0.592722	1.953952	0	0	0	0	43
■	64	same_day_direction_start_15_ total2	The number of requests that are made with a request interval equal to or larger as 15 minutes on the same operationdate from this stop cluster with the same line and in the same direction (excluding the current trip).	97825		50.2665	113.0371	0	1	14	49	1503

	Feature	Description	count	unique	mean	std	min	0.25	0.5	0.75	max
65	same_day_direction_start_15_ total2_3hour	The number of requests that are made with a request interval equal to or larger as 15 minutes on the same operationdate from this stop cluster with the same line and in the same direction (excluding the current trip) within an interval of 3 hours of the planned departure time.	97825		17.21341	40.40924	0	0	4	16	526
66	same_day_direction_start_15_ total2_3hour_before	The number of requests that are made with a request interval equal to or larger as 15 minutes on the same operationdate from this stop cluster with the same line and in the same direction (excluding the current trip) only including the trips which are 3 hours in advance of the planned departure time.	97825		12.42729	30.67879	0	0	3	12	442
67	cko_no	The number of registered check outs.	97825		1.006154	2.536079	0	0	0	1	51
68	cko_no_historic_avg	The mean number of check outs based on the historic trips.	97825		0.988633	2.143371	0	0	0	1	27
69	end_15_total_historic_avg	The mean number of requests ending at this stop and having a request interval larger than 14 minutes based on historic trips.	97825		0.682249	2.036817	0	0	0	1	38
70	historic_residual_end_15_total	The difference between the historic mean and the current total requests ending at this stop (end_15_total_historic_avg minus end_15_total).	97825		-0.07702	1.199162	-18	0	0	0	31
71	requests_per_cko_historic_avg	The historic ratio between requests and check ins.	97825		68.27317	121.8074	0	1	1	125	1000
72	end_15_total	The number of requests ending at stop i and having a request interval larger than 14 minutes.	97825		0.605234	2.024165	0	0	0	0	49
73	end_15	The number of requests ending at stop i and having a request interval between 15 and 29 minutes.	97825		0.072568	0.349655	0	0	0	0	12

		Feature	Description	count	unique	mean	std	min	0.25	0.5	0.75	max
■	74	end_30	The number of requests ending at stop i and having a request interval between 30 and 59 minutes.	97825		0.09424	0.431217	0	0	0	0	15
■	75	end_60	The number of requests ending at stop i and having a request interval between 60 and 119 minutes.	97825		0.087708	0.413986	0	0	0	0	19
■	76	end_120	The number of requests ending at stop i and having a request interval between 120 and 359 minutes.	97825		0.088249	0.425597	0	0	0	0	15
■	77	end_360	The number of requests ending at stop i and having a request interval between 360 and 1440 minutes.	97825		0.192538	0.986738	0	0	0	0	33
■	78	end_1440	The number of requests ending at stop i and having a request interval between 1440 and 2879 minutes.	97825		0.026578	0.235012	0	0	0	0	24
■	79	end_2880	The number of requests ending at stop i and having a request interval larger as 2879 minutes.	97825		0.043353	0.350229	0	0	0	0	20
▨	80	end_15_j_walking_0	The number of requests ending at stop i with a walking time smaller than 20 minutes in the total travel advice.	97825		0.585883	1.968836	0	0	0	0	49
▨	81	end_15_j_walking_20	The number of requests ending at stop i with a walking time larger than 19 minutes in the total travel advice.	97825		0.019351	0.197671	0	0	0	0	12
▨	82	end_15_j_firstlegqbuzznot	The number of requests ending at stop i and where the request is not the first transit part in the total travel advice.	97825		0.206174	0.863807	0	0	0	0	28
▨	83	end_15_j_firstlegqbuzz	The number of requests ending at stop i and where the request is the first transit part in the total travel advice.	97825		0.39906	1.575736	0	0	0	0	41
▨	84	end_15_j_multiple_60not	The number of requests ending at stop i for which the request interval of the total travel advice is not a multiple of 60 minutes.	97825		0.529486	1.848747	0	0	0	0	47
▨	85	end_15_j_multiple_60	The number of requests ending at stop i for which the request interval of the	97825		0.075748	0.37503	0	0	0	0	14

	Feature	Description	count	unique	mean	std	min	0.25	0.5	0.75	max
		total travel advice is a multiple of 60 minutes.									
86	end_15_j_transittransfer0	The number of requests ending at stop i which have 1 transit part in the total travel advice.	97825		0.255937	0.891817	0	0	0	0	24
87	end_15_j_transittransfer1	The number of requests ending at stop i which have 2 transit parts in the total travel advice.	97825		0.240337	1.075982	0	0	0	0	34
88	end_15_j_transittransfer2	The number of requests ending at stop i which have 3 transit parts in the total travel advice.	97825		0.075257	0.448254	0	0	0	0	21
89	end_15_j_transittransfer3	The number of requests ending at stop i which have 4 transit parts in the total travel advice.	97825		0.025178	0.248832	0	0	0	0	14
90	end_15_j_transittransfer3morethan	The number of requests ending at stop i which have more than 4 transit parts in the total travel advice.	97825		0.008525	0.139764	0	0	0	0	13
91	end_15_j_speed50smallerthan_ distance100morethan_not	The number of requests ending at stop i for which the total travel advice has not a Euclidean speed smaller than 50 and a Euclidian distance equal to or more than 100 km.	97825		0.597996	1.992946	0	0	0	0	49
92	end_15_j_speed50smallerthan_ distance100morethan	The number of requests ending at stop i for which the total travel advice has a Euclidean speed smaller than 50 and a Euclidian distance equal to or more than 100 km.	97825		0.007237	0.14042	0	0	0	0	15
93	end_15_j_speed0biggerthan	The number of requests ending at stop i for which the speed of the total travel advice is smaller than 10.	97825		0.077393	0.364362	0	0	0	0	10
94	end_15_j_speed10biggerthan	The number of requests ending at stop i for which the speed of the total travel advice is bigger than 9.	97825		0.527841	1.919521	0	0	0	0	48
95	end_15_p_traveltime0biggerthan	The number of requests ending at stop i and having a travel time between 0 and 9 minutes.	97825		0.169088	0.699521	0	0	0	0	25

	Feature	Description	count	unique	mean	std	min	0.25	0.5	0.75	max
96	end_15_p_traveltime10biggerthan	The number of requests ending at stop i and having a travel time between 10 and 19 minutes.	97825		0.235032	1.028066	0	0	0	0	34
97	end_15_p_traveltime20biggerthan	The number of requests ending at stop i and having a travel time between 20 and 29 minutes.	97825		0.133902	0.721596	0	0	0	0	27
98	end_15_p_traveltime30biggerthan	The number of requests ending at stop i and having a travel time larger as 29 minutes.	97825		0.067212	0.504144	0	0	0	0	25
99	prev_trip_end_15_total	The number of requests from the previous trip ending at stop i and having a travel time larger as 15 minutes.	97825		0.59493	2.012358	0	0	0	0	49
100	next_trip_end_15_total	The number of requests from the next trip ending at stop i and having a travel time larger as 15 minutes.	97825		0.600542	2.01078	0	0	0	0	49
101	prev_stop_end_15_total	The number of requests ending at stop i-1 and having a travel time larger as 15 minutes.	97825		0.597996	2.022346	0	0	0	0	49
102	next_stop_end_15_total	The number of requests ending at stop i+1 and having a travel time larger as 15 minutes.	97825		0.605234	2.024165	0	0	0	0	49
103	same_day_direction_end_15_ total2	The number of requests that are made with a request interval equal to or larger as 15 minutes on the same operationdate from this stop cluster with the same line and in the same direction (excluding the current trip).	97825		51.13182	138.0933	0	0	9	44	1453
104	same_day_direction_end_15_ total2_3hour	The number of requests that are made with a request interval equal to or larger as 15 minutes on the same operationdate from this stop cluster with the same line and in the same direction (excluding the current trip) within an interval of 3 hours of the planned departure time.	97825		18.02089	48.28955	0	0	3	16	661
105	same_day_direction_end_15_ total2_3hour_before	The number of requests that are made with a request interval equal to or larger	97825		12.75176	35.4939	0	0	2	11	557

Feature	Description	count	unique	mean	std	min	0.25	0.5	0.75	max
[redacted]	as 15 minutes on the same operationdate from this stop cluster with the same line and in the same direction (excluding the current trip) only including the trips which are 3 hours in advance of the planned departure time.									

Appendix O Possible line configurations for trips around 8 AM

Line	Configuration	Number of trips	Number of records	Headway	Standard deviation headway
g554	g554-1-0	239	10277	10.05546366	0.323947786
g554	g554-7-1	235	9870	10.07720365	0.37165379
g505	g505-5-1	227	6356	10.00283197	0.773066383
g503	g503-1-0	120	4680	11.01282051	3.132031931
g503	g503-5-1	120	4680	10.52564103	3.155407774
g503	g503-6-1	120	4680	11.02564103	3.034211253
g503	g503-2-0	119	4641	11.2693385	3.064441744
g078	g078-5-1	55	4180	39.61356297	10.27169373
g556	g556-2-1	134	4154	10.22628792	1.697669779
g505	g505-1-0	148	4144	13.14454633	6.647469647
g552	g552-3-1	111	4107	15	0
g552	g552-1-0	114	3762	15	0
g065	g065-4-1	89	3728	22.33449571	7.388667162
g556	g556-1-1	80	3520	17.80085227	8.833938879
g559	g559-6-1	60	3480	25.75862069	6.877510859
g556	g556-1-0	77	3388	17.54309327	8.731921023
g078	g078-1-0	54	3294	54.52884032	9.20503632
g559	g559-1-0	55	3245	21.25423729	7.396954547
g556	g556-2-0	82	2542	11.99252557	2.873624304
g300	g300-3-1	60	2520	30	0
g065	g065-1-0	61	2512	30.3125	1.204942612
g061	g061-6-0	29	2465	53.57317073	11.3441177
g061	g061-2-0	28	2380	98.64705882	117.4370401
g061	g061-2-1	28	2380	50.78823529	12.80351187
g557	g557-4-1	82	2214	15	0
g078	g078-6-1	28	2156	39.93333333	0.249503238
g178	g178-3-0	51	2091	39.10186514	9.681959318
g559	g559-3-0	58	2030	22.25615764	8.36473853
g300	g300-1-1	235	1880	9.956914894	0.674759771
g559	g559-7-1	27	1728	45.703125	116.9081881
g061	g061-15-1	23	1679	50.32876712	21.57381265
g300	g300-1-0	205	1640	9.823170732	0.567967117
g551	g551-2-1	60	1500	19.2	6.737229414
g061	g061-1-1	29	1421	33.44897959	25.49531234
g557	g557-2-0	48	1344	15	0
g565	g565-1-0	146	1314	5	0
d032	d032-2-0	30	1260	24	0
g557	g557-1-0	54	1242	17.7173913	18.96716286
d032	d032-1-1	30	1140	58.50877193	6.331832998

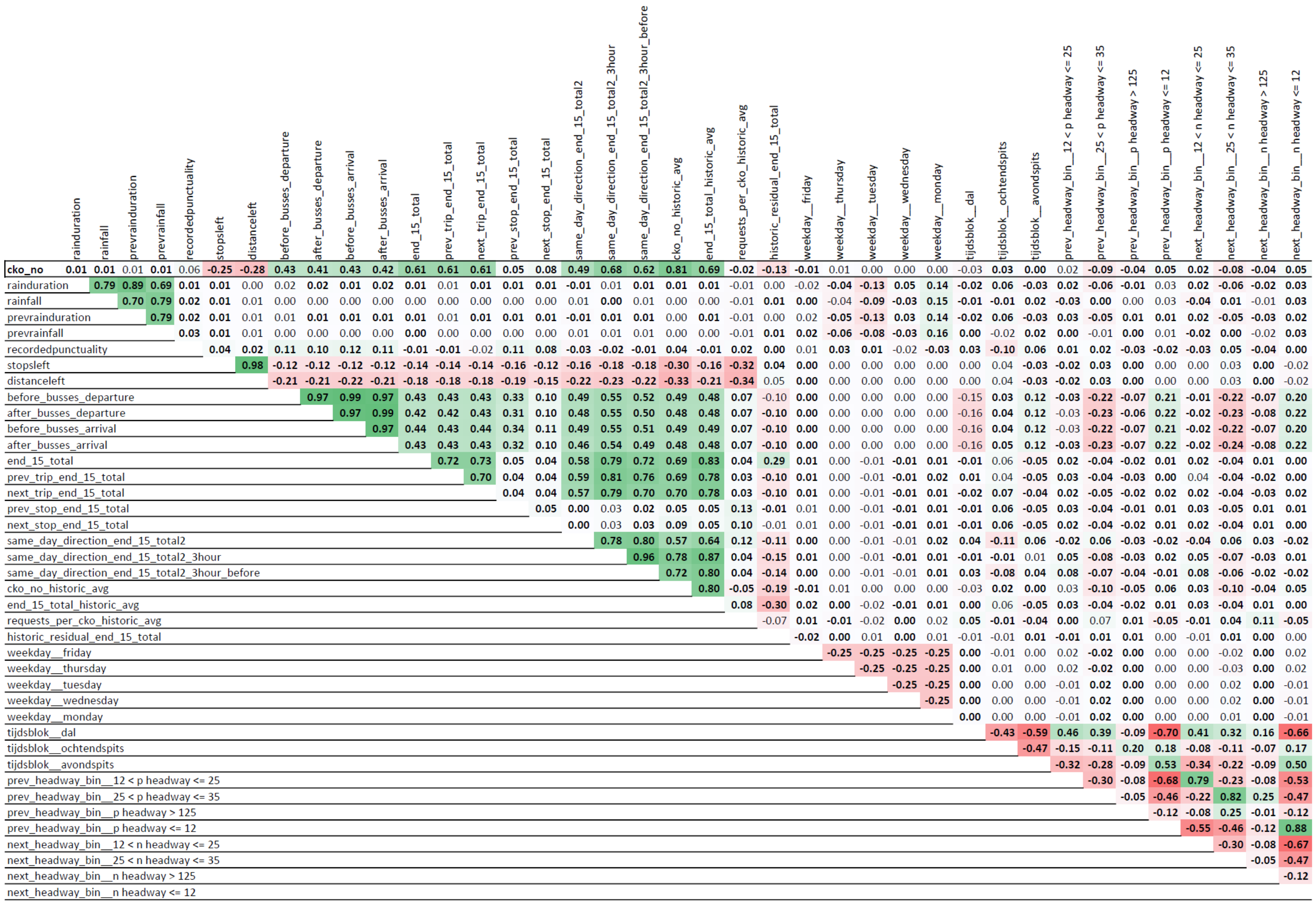
Line	Configuration	Number of trips	Number of records	Headway	Standard deviation headway
d032	d032-1-0	30	1140	60	0
g559	g559-5-0	29	1102	14.27306617	3.22261592
g559	g559-8-1	28	1008	23.5	7.277995293
g565	g565-1-1	112	1008	7.455357143	2.500842176
d031	d031-1-0	30	990	60	0
g061	g061-3-0	29	928	44.50323276	7.507923919
g564	g564-1-0	60	900	30	0
d131	d131-1-1	30	840	60	0
g505	g505-2-0	58	812	16.07142857	6.179383521
g551	g551-2-0	30	750	19.368	6.713121281
g061	g061-7-0	21	630	455.5666667	0.495929376
g551	g551-1-0	30	570	15.55438596	3.560379029
g551	g551-1-1	30	570	15.78947368	3.352395162
g012	g012-5-1	21	556	60	0
g564	g564-2-1	30	510	30	0
d039	d039-1-0	27	378	60	0
d039	d039-1-1	27	378	60	0

Appendix P Correlation matrix of the boarding model for in g554-1-0 during Weekdays

	rainduration	rainfall	prevrainduration	prevrainfall	recordedpunctuality	stopsleft	distanceleft	before_busses_departure	after_busses_departure	before_busses_arrival	after_busses_arrival	start_15_total	prev_trip_start_15_total	next_trip_start_15_total	prev_stop_start_15_total	next_stop_start_15_total	same_day_direction_start_15_total2	same_day_direction_start_15_total2_3hour	same_day_direction_start_15_total2_3hour_before	cki_no_historic_avg	start_15_total_historic_avg	requests_per_cki_historic_avg	historic_residual_start_15_total	weekday__friday	weekday__thursday	weekday__tuesday	weekday__wednesday	weekday__monday	tijdsblok__dal	tijdsblok__ochtendspits	tijdsblok__avondspits	prev_headway_bin__p headway <= 12	prev_headway_bin__12 < p headway <= 25	prev_headway_bin__25 < p headway <= 35	prev_headway_bin__p headway > 125	next_headway_bin__n headway <= 12	next_headway_bin__12 < n headway <= 25	next_headway_bin__25 < n headway <= 35	next_headway_bin__n headway > 125
cki_no	**0.01**	**0.01**	0.01	**0.01**	**-0.01**	**0.06**	**0.02**	**0.47**	**0.48**	**0.49**	**0.47**	**0.62**	**0.60**	**0.61**	**0.02**	**0.00**	0.54	**0.70**	**0.64**	**0.83**	**0.71**	**-0.06**	**-0.17**	**-0.01**	0.01	0.00	0.00	0.00	-0.03	**0.04**	**-0.01**	**0.04**	0.01	**-0.06**	**-0.03**	**0.03**	**0.02**	**-0.06**	**-0.03**
rainduration		**0.79**	**0.89**	**0.69**	**0.01**	**0.01**	0.00	0.02	**0.02**	**0.01**	**0.02**	**0.01**	0.01	**0.02**	**0.01**	**0.01**	**-0.01**	0.01	**0.01**	**0.01**	0.01	**0.00**	0.00	-0.02	**-0.04**	**-0.13**	**0.05**	**0.14**	**-0.02**	**0.06**	**-0.03**	0.03	**0.02**	**-0.06**	**-0.01**	**0.03**	**0.02**	**-0.06**	**-0.02**
rainfall			**0.70**	**0.79**	**0.02**	**0.01**	0.01	0.00	0.00	0.00	0.00	0.01	**0.00**	0.01	0.00	0.01	0.01	**0.01**	0.01	0.00	0.00	0.00	0.01	**0.00**	-0.04	**-0.09**	**-0.03**	**0.15**	**-0.01**	**-0.01**	**0.02**	0.03	**-0.03**	**0.00**	0.00	**0.03**	**-0.04**	**0.01**	-0.01
prevrainduration				**0.79**	**0.02**	**0.01**	0.01	0.01	**0.01**	**0.01**	**0.01**	**0.01**	0.01	**0.01**	**0.01**	**0.01**	**-0.01**	**0.01**	**0.01**	0.01	0.01	0.00	**0.00**	0.02	**-0.05**	**-0.13**	**0.03**	**0.14**	**-0.02**	**0.06**	**-0.03**	**0.01**	**0.03**	**-0.05**	**0.01**	**0.02**	**0.02**	**-0.05**	**-0.03**
prevrainfall					**0.03**	**0.01**	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01	**0.01**	**0.01**	0.00	0.00	0.00	**0.01**	**0.02**	**-0.06**	**-0.08**	**-0.03**	**0.16**	**0.00**	-0.02	**0.02**	0.01	**0.00**	-0.01	**0.00**	**0.03**	**-0.02**	**0.00**	-0.02
recordedpunctuality						**0.04**	**0.02**	**0.11**	**0.10**	**0.12**	**0.11**	**-0.05**	**-0.05**	**-0.05**	**0.11**	**0.02**	**-0.07**	**-0.06**	**-0.05**	-0.02	**-0.07**	**0.05**	**0.03**	0.01	**0.03**	**0.01**	-0.02	**-0.03**	**0.03**	**-0.10**	**0.06**	**-0.02**	**0.01**	**0.02**	**-0.03**	**0.00**	**-0.03**	**0.05**	**-0.04**
stopsleft							**0.98**	**-0.12**	**-0.12**	**-0.12**	**-0.12**	**0.12**	**0.12**	**0.12**	**0.09**	**0.13**	**0.20**	**0.16**	**0.15**	**0.08**	**0.12**	**0.35**	**-0.02**	**0.00**	0.00	0.00	0.00	0.00	0.00	0.04	**-0.03**	0.00	**-0.02**	**0.03**	**0.00**	-0.02	0.00	0.03	**0.00**
distanceleft								**-0.21**	**-0.21**	**-0.22**	**-0.21**	**0.08**	**0.09**	**0.08**	**0.06**	**0.11**	**0.16**	**0.12**	**0.11**	**0.03**	**0.08**	**0.34**	**0.00**	**0.00**	0.00	0.00	0.00	0.00	0.00	0.04	**-0.03**	0.00	**-0.02**	**0.03**	**0.00**	-0.02	0.00	0.03	**0.00**
before_busses_departure									**0.97**	**0.99**	**0.97**	**0.37**	**0.37**	**0.38**	**0.29**	**0.02**	**0.37**	**0.49**	**0.45**	**0.54**	**0.44**	**-0.03**	**-0.12**	**0.00**	0.00	0.00	0.00	0.00	-0.15	**0.03**	**0.12**	**0.21**	**-0.03**	**-0.22**	**-0.07**	**0.20**	**-0.01**	**-0.22**	**-0.07**
after_busses_departure										**0.97**	**0.99**	**0.37**	**0.36**	**0.38**	**0.28**	**0.01**	**0.37**	**0.49**	**0.44**	**0.55**	**0.44**	**-0.04**	**-0.12**	**0.00**	0.00	0.00	0.00	0.00	-0.16	**0.04**	**0.12**	**0.22**	-0.03	**-0.23**	**-0.06**	**0.22**	**-0.02**	**-0.23**	**-0.08**
before_busses_arrival											**0.97**	**0.38**	**0.38**	**0.39**	**0.29**	**0.02**	**0.37**	**0.49**	**0.45**	**0.55**	**0.45**	**-0.04**	**-0.12**	**0.00**	0.00	0.00	0.00	0.00	-0.16	**0.04**	**0.12**	**0.21**	-0.03	**-0.22**	**-0.07**	**0.20**	**-0.02**	**-0.22**	**-0.07**
after_busses_arrival												**0.36**	**0.36**	**0.38**	**0.28**	**0.02**	**0.34**	**0.47**	**0.43**	**0.54**	**0.43**	**-0.04**	**-0.12**	**0.00**	0.00	0.00	0.00	0.00	-0.16	**0.05**	**0.12**	**0.22**	**-0.03**	**-0.23**	**-0.07**	**0.22**	**-0.02**	**-0.24**	**-0.08**
start_15_total													**0.70**	**0.69**	**0.01**	**0.02**	**0.63**	**0.75**	**0.70**	**0.67**	**0.81**	**0.02**	**0.28**	**0.01**	0.00	-0.01	**-0.01**	**0.02**	**-0.02**	**0.08**	**-0.05**	**0.01**	**0.01**	**-0.03**	**-0.02**	**-0.01**	**0.03**	**-0.03**	**0.01**
prev_trip_start_15_total														**0.67**	**0.01**	**0.01**	**0.63**	**0.78**	**0.74**	**0.65**	**0.74**	**0.01**	**-0.08**	**0.01**	0.00	-0.01	**-0.01**	**0.02**	**-0.01**	**0.07**	**-0.05**	**0.00**	**0.02**	**-0.03**	**-0.04**	**-0.01**	0.04	**-0.03**	**-0.02**
next_trip_start_15_total															**0.01**	**0.01**	**0.62**	**0.76**	**0.68**	**0.66**	**0.73**	**0.01**	**-0.09**	**0.01**	0.00	-0.01	**-0.01**	**0.01**	**-0.03**	**0.09**	**-0.05**	**0.02**	**0.01**	**-0.04**	**-0.02**	**0.01**	**0.02**	**-0.03**	**-0.04**
prev_stop_start_15_total																**0.03**	**-0.02**	**0.01**	0.01	**0.02**	**0.00**	**0.08**	**0.01**	**0.01**	0.00	-0.01	**-0.01**	**0.02**	**-0.02**	**0.08**	**-0.05**	**0.01**	**0.02**	**-0.04**	**-0.02**	**0.00**	**0.03**	**-0.04**	**0.01**
next_stop_start_15_total																	**0.00**	0.01	**0.02**	**0.01**	**0.02**	**0.12**	-0.01	**0.01**	0.00	-0.01	**-0.01**	**0.01**	**-0.02**	**0.08**	**-0.05**	**0.01**	**0.01**	**-0.02**	**-0.02**	**0.00**	**0.02**	**-0.03**	**0.01**
same_day_direction_start_15_total2																		**0.82**	**0.82**	**0.61**	**0.69**	**0.10**	-0.12	**0.00**	0.00	-0.01	-0.02	**0.02**	**0.05**	**-0.13**	**0.07**	**-0.03**	**-0.04**	**0.10**	**-0.03**	**-0.04**	**-0.04**	**0.09**	**0.04**
same_day_direction_start_15_total2_3hour																			**0.97**	**0.78**	**0.83**	**0.03**	**-0.14**	**0.01**	**0.00**	-0.01	**-0.02**	**0.02**	**-0.02**	**0.02**	**0.00**	**0.01**	0.05	**-0.08**	**-0.03**	**0.00**	**0.06**	**-0.07**	**-0.03**
same_day_direction_start_15_total2_3hour_before																				**0.71**	**0.77**	**0.02**	**-0.13**	**0.00**	0.00	-0.01	**-0.01**	**0.02**	**0.02**	**-0.06**	**0.03**	**-0.02**	**0.08**	**-0.07**	**-0.05**	**-0.04**	**0.09**	**-0.06**	**-0.03**
cki_no_historic_avg																					**0.80**	**-0.09**	**-0.23**	**0.00**	0.01	0.00	0.00	0.00	-0.04	**0.04**	**0.00**	**0.05**	0.02	**-0.07**	**-0.04**	**0.04**	**0.02**	**-0.07**	**-0.03**
start_15_total_historic_avg																						**0.07**	**-0.34**	0.02	**0.00**	-0.01	**-0.01**	**0.01**	**-0.02**	**0.09**	**-0.06**	**0.01**	**0.02**	**-0.03**	**-0.02**	**-0.01**	**0.03**	**-0.03**	**0.01**
requests_per_cki_historic_avg																							**-0.08**	0.03	**-0.01**	**-0.01**	**-0.01**	**0.00**	0.01	**0.03**	**-0.04**	**-0.01**	**-0.03**	**0.03**	**0.05**	**-0.02**	**-0.02**	**0.04**	**0.04**
historic_residual_start_15_total																								**-0.02**	**0.00**	0.01	0.00	0.01	0.00	-0.02	**0.02**	0.00	**-0.01**	**0.01**	**0.00**	0.01	-0.01	**0.01**	0.00
weekday__friday																									**-0.25**	**-0.25**	**-0.25**	**-0.25**	**0.00**	-0.01	0.00	0.00	0.02	**-0.02**	**0.00**	0.02	0.00	-0.02	**0.00**
weekday__thursday																										**-0.25**	**-0.25**	**-0.25**	**0.00**	0.01	0.00	0.00	0.02	**-0.02**	**0.00**	0.02	0.00	-0.03	**0.00**
weekday__tuesday																											**-0.25**	**-0.25**	**0.00**	0.00	0.00	0.00	-0.01	**0.02**	**0.00**	-0.01	0.00	0.02	**0.00**
weekday__wednesday																												**-0.25**	**0.00**	0.00	0.00	0.00	-0.01	**0.02**	**0.00**	-0.01	0.00	0.02	**0.00**
weekday__monday																													**0.00**	0.00	0.00	0.00	-0.01	**0.02**	**0.00**	-0.01	0.00	0.01	**0.00**
tijdsblok__dal																														**-0.43**	**-0.59**	**-0.70**	**0.46**	**0.39**	**-0.09**	**-0.66**	**0.41**	**0.32**	**0.16**
tijdsblok__ochtendspits																															**-0.47**	**0.18**	**-0.15**	**-0.11**	**0.20**	**0.17**	**-0.08**	**-0.11**	**-0.07**
tijdsblok__avondspits																																**0.53**	**-0.32**	**-0.28**	**-0.09**	**0.50**	**-0.34**	**-0.22**	**-0.09**
prev_headway_bin__12 < p headway <= 25																																	**-0.68**	**-0.30**	**-0.08**	**-0.53**	**0.79**	**-0.23**	**-0.08**
prev_headway_bin__25 < p headway <= 35																																		**-0.46**	**-0.05**	**-0.47**	**-0.22**	**0.82**	**0.25**
prev_headway_bin__p headway > 125																																			**-0.12**	**-0.12**	**-0.08**	**0.25**	**-0.01**
prev_headway_bin__p headway <= 12																																				**0.88**	**-0.55**	**-0.46**	**-0.12**
next_headway_bin__12 < n headway <= 25																																					**-0.67**	**-0.30**	**-0.08**
next_headway_bin__25 < n headway <= 35																																						**-0.47**	**-0.05**
next_headway_bin__n headway > 125																																							**-0.12**

Figure P.1: Correlation matrix of the features for the barding model of the g554-1-0/Workdays partition. The Pearson's r values in bold are significant with p =< 0.0.1

Appendix Q Correlation matrix of the alighting model for g554-1-0 during Weekdays



	rainduration	rainfall	prevrainduration	prevrainfall	recordedpunctuality	stopsleft	distanceleft	before_busses_departure	after_busses_departure	before_busses_arrival	after_busses_arrival	end_15_total	prev_trip_end_15_total	next_trip_end_15_total	prev_stop_end_15_total	next_stop_end_15_total	same_day_direction_end_15_total2	same_day_direction_end_15_total2_3hour	same_day_direction_end_15_total2_3hour_before	cko_no_historic_avg	end_15_total_historic_avg	requests_per_cko_historic_avg	historic_residual_end_15_total	weekday__friday	weekday__thursday	weekday__tuesday	weekday__wednesday	weekday__monday	tijdsblok__dal	tijdsblok__ochtendspits	tijdsblok__avondspits	prev_headway_bin__12 < p headway <= 25	prev_headway_bin__25 < p headway <= 35	prev_headway_bin__p headway > 125	prev_headway_bin__p headway <= 12	next_headway_bin__12 < n headway <= 25	next_headway_bin__25 < n headway <= 35	next_headway_bin__n headway > 125	next_headway_bin__n headway <= 12
cko_no	0.01	0.01	0.01	0.01	0.06	-0.25	-0.28	0.43	0.41	0.43	0.42	0.61	0.61	0.61	0.05	0.08	0.49	0.68	0.62	0.81	0.69	-0.02	-0.13	-0.01	0.01	0.00	0.00	0.00	-0.03	0.03	0.00	0.02	-0.09	-0.04	0.05	0.02	-0.08	-0.04	0.05
rainduration		0.79	0.89	0.69	0.01	0.01	0.00	0.02	0.02	0.01	0.02	0.01	0.01	0.01	0.01	0.01	-0.01	0.01	0.01	0.01	0.01	-0.01	0.00	-0.02	-0.04	-0.13	0.05	0.14	-0.02	0.06	-0.03	0.02	-0.06	-0.01	0.03	0.02	-0.06	-0.02	0.03
rainfall			0.70	0.79	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	-0.01	0.01	0.00	-0.04	-0.09	-0.03	0.15	-0.01	-0.01	0.02	-0.03	0.00	0.00	0.03	-0.04	0.01	-0.01	0.03
prevrainduration				0.79	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	-0.01	0.01	0.01	0.00	0.01	-0.01	0.00	0.02	-0.05	-0.13	0.03	0.14	-0.02	0.06	-0.03	0.03	-0.05	0.01	0.01	0.02	-0.05	-0.03	0.02
prevrainfall					0.03	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	-0.01	0.01	0.02	-0.06	-0.08	-0.03	0.16	0.00	-0.02	0.02	0.00	-0.01	0.00	0.01	-0.02	0.00	-0.02	0.03
recordedpunctuality						0.04	0.02	0.11	0.10	0.12	0.11	-0.01	-0.01	-0.02	0.11	0.08	-0.03	-0.02	-0.01	0.04	-0.01	0.02	0.00	0.01	0.03	0.01	-0.02	-0.03	0.03	-0.10	0.06	0.01	0.02	-0.03	-0.02	-0.03	0.05	-0.04	0.00
stopsleft							0.98	-0.12	-0.12	-0.12	-0.12	-0.14	-0.14	-0.14	-0.16	-0.12	-0.16	-0.18	-0.18	-0.30	-0.16	-0.32	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.04	-0.03	-0.02	0.03	0.00	0.00	0.00	0.03	0.00	-0.02
distanceleft								-0.21	-0.21	-0.22	-0.21	-0.18	-0.18	-0.18	-0.19	-0.15	-0.22	-0.23	-0.22	-0.33	-0.21	-0.34	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.04	-0.03	-0.02	0.03	0.00	0.00	0.00	0.03	0.00	-0.02
before_busses_departure									0.97	0.99	0.97	0.43	0.43	0.43	0.33	0.10	0.49	0.55	0.52	0.49	0.48	0.07	-0.10	0.00	0.00	0.00	0.00	0.00	-0.15	0.03	0.12	-0.03	-0.22	-0.07	0.21	-0.01	-0.22	-0.07	0.20
after_busses_departure										0.97	0.99	0.42	0.42	0.43	0.31	0.10	0.48	0.55	0.50	0.48	0.48	0.07	-0.10	0.00	0.00	0.00	0.00	0.00	-0.16	0.04	0.12	-0.03	-0.23	-0.06	0.22	-0.02	-0.23	-0.08	0.22
before_busses_arrival											0.97	0.44	0.43	0.44	0.34	0.11	0.49	0.55	0.51	0.49	0.49	0.07	-0.10	0.00	0.00	0.00	0.00	0.00	-0.16	0.04	0.12	-0.03	-0.22	-0.07	0.21	-0.02	-0.22	-0.07	0.20
after_busses_arrival												0.43	0.43	0.43	0.32	0.10	0.46	0.54	0.49	0.48	0.48	0.07	-0.10	0.00	0.00	0.00	0.00	0.00	-0.16	0.05	0.12	-0.03	-0.23	-0.07	0.22	-0.02	-0.24	-0.08	0.22
end_15_total													0.72	0.73	0.05	0.04	0.58	0.79	0.72	0.69	0.83	0.04	0.29	0.01	0.00	-0.01	-0.01	0.01	-0.01	0.06	-0.05	0.02	-0.04	-0.02	0.01	0.02	-0.04	0.01	0.00
prev_trip_end_15_total														0.70	0.04	0.04	0.59	0.81	0.76	0.69	0.78	0.03	-0.10	0.01	0.00	-0.01	-0.01	0.02	0.01	0.04	-0.05	0.03	-0.04	-0.03	0.00	0.04	-0.04	-0.02	0.00
next_trip_end_15_total															0.04	0.04	0.57	0.79	0.70	0.70	0.78	0.03	-0.10	0.01	0.00	-0.01	-0.01	0.01	-0.02	0.07	-0.04	0.02	-0.05	-0.02	0.02	0.02	-0.04	-0.03	0.02
prev_stop_end_15_total																0.05	0.00	0.03	0.02	0.05	0.05	0.13	-0.01	0.01	0.00	-0.01	-0.01	0.01	-0.01	0.06	-0.05	0.03	-0.04	-0.01	0.01	0.03	-0.05	0.01	0.01
next_stop_end_15_total																	0.00	0.03	0.03	0.09	0.05	0.10	-0.01	0.01	0.00	-0.01	-0.01	0.01	-0.01	0.06	-0.05	0.02	-0.04	-0.02	0.01	0.02	-0.04	0.01	0.00
same_day_direction_end_15_total2																		0.78	0.80	0.57	0.64	0.12	-0.11	0.00	0.00	-0.01	-0.01	0.02	0.04	-0.11	0.06	-0.02	0.06	-0.03	-0.02	-0.04	0.06	0.03	-0.02
same_day_direction_end_15_total2_3hour																			0.96	0.78	0.87	0.04	-0.15	0.01	0.00	-0.01	-0.01	0.01	-0.01	-0.01	0.01	0.05	-0.08	-0.03	0.02	0.05	-0.07	-0.03	0.01
same_day_direction_end_15_total2_3hour_before																				0.72	0.80	0.04	-0.14	0.00	0.00	-0.01	-0.01	0.01	0.03	-0.08	0.04	0.08	-0.07	-0.04	-0.01	0.08	-0.06	-0.02	-0.02
cko_no_historic_avg																					0.80	-0.05	-0.19	-0.01	0.01	0.00	0.00	0.00	-0.03	0.02	0.00	0.03	-0.10	-0.05	0.06	0.03	-0.10	-0.04	0.05
end_15_total_historic_avg																						0.08	-0.30	0.02	0.00	-0.02	-0.01	0.01	0.00	0.06	-0.05	0.03	-0.04	-0.02	0.01	0.03	-0.04	0.01	0.00
requests_per_cko_historic_avg																							-0.07	0.01	-0.01	-0.02	0.00	0.02	0.05	-0.01	-0.04	0.00	0.07	0.01	-0.05	-0.01	0.04	0.11	-0.05
historic_residual_end_15_total																								-0.02	0.00	0.01	0.00	0.01	-0.01	-0.01	0.01	-0.01	0.01	0.01	0.00	-0.01	0.01	0.00	0.00
weekday__friday																									-0.25	-0.25	-0.25	-0.25	0.00	-0.01	0.00	0.02	-0.02	0.00	0.00	0.00	-0.02	0.00	0.02
weekday__thursday																										-0.25	-0.25	-0.25	0.00	0.01	0.00	0.02	-0.02	0.00	0.00	0.00	-0.03	0.00	0.02
weekday__tuesday																											-0.25	-0.25	0.00	0.00	0.00	-0.01	0.02	0.00	0.00	0.00	0.02	0.00	-0.01
weekday__wednesday																												-0.25	0.00	0.00	0.00	-0.01	0.02	0.00	0.00	0.00	0.02	0.00	-0.01
weekday__monday																													0.00	0.00	0.00	-0.01	0.02	0.00	0.00	0.00	0.01	0.00	-0.01
tijdsblok__dal																														-0.43	-0.59	0.46	0.39	-0.09	-0.70	0.41	0.32	0.16	-0.66
tijdsblok__ochtendspits																															-0.47	-0.15	-0.11	0.20	0.18	-0.08	-0.11	-0.07	0.17
tijdsblok__avondspits																																-0.32	-0.28	-0.09	0.53	-0.34	-0.22	-0.09	0.50
prev_headway_bin__12 < p headway <= 25																																	-0.30	-0.08	-0.68	0.79	-0.23	-0.08	-0.53
prev_headway_bin__25 < p headway <= 35																																		-0.05	-0.46	-0.22	0.82	0.25	-0.47
prev_headway_bin__p headway > 125																																			-0.12	-0.08	0.25	-0.01	-0.12
prev_headway_bin__p headway <= 12																																				-0.55	-0.46	-0.12	0.88
next_headway_bin__12 < n headway <= 25																																					-0.30	-0.08	-0.67
next_headway_bin__25 < n headway <= 35																																						-0.05	-0.47
next_headway_bin__n headway > 125																																							-0.12
next_headway_bin__n headway <= 12																																							

Figure Q.1: Correlation matrix of the features for the alighting model of the g554-1-0/Workdays partition. The Pearson's r values in bold are significant with p =< 0.0.1

Appendix R String diagram

The data in the *bus data*-dataset for one line can be visualized using a string diagram. Figure R.1 visualizes the difference between the scheduled bus passage times and the actual times. Each dot on the line represents the stopping at (or passing of) a stop. Lines with a positive slope start in Groningen and end in Opende and vice versa. If the slope is bigger the bus has more speed. The diagram also shows also that the stops are not always evenly spaced. Furthermore, at some stops the line is horizontal, like at stop *Zuidhorn, Station*. Such stops are called timing stops, which are used to make transfers possible. For example, stop *Zuidhorn, Station* is located near a train station where trains stop on the hour and on the half hour. These timing stops are also useful to make up for some delays.

Each line color denotes a different bus. Thus, most of the time when a bus finishes a trip, it continues with the next trip in opposite direction. From the diagram, we can deduct that the headway between the buses of this line is 1 hour (thus a frequency of 1 bus per hour); an hour passes before the next trip in the same direction starts. Also, these lines show that the buses of trips 1010 and 1012 serviced the stops mostly ahead of schedule and that the buses of trip 1014 and 1023 mostly are lagging behind. It can be seen that the timing stop resets the delay. Because buses service consecutive trips, it could be that a delay is inherited from a previous trip, however this does not occur in Figure R.1**Error! Reference source not found.**. This phenomenon only occurs in 14091 trips in the whole period.

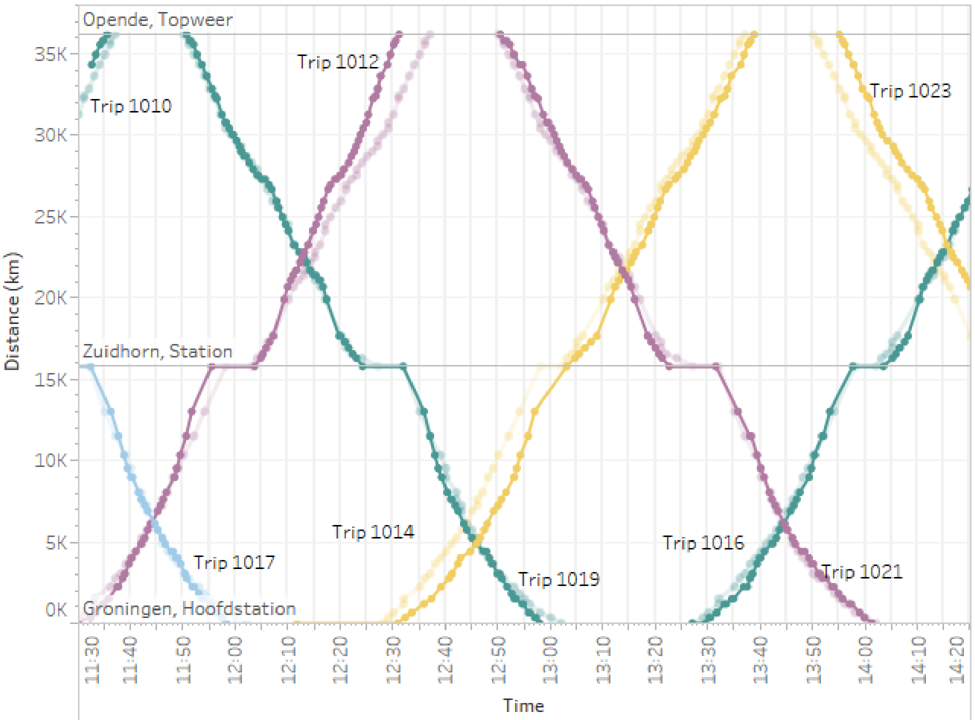


Figure R.1: The space time diagram for the trips with lineplanningnumber g039 - Surhuisterveen - Groningen on Monday 09-01-2017 between 11:30 and 14:30. The solid lines are the actual bus passage times and the transparent are the scheduled variant. Each line color denotes a particular bus.

Appendix S Decision tree forecasting model example

This model for forecasting the number of people boarding is trained with data from partition 1. The decision tree has a max depth of 4 in order to visualize it. The scores are not cross validated.

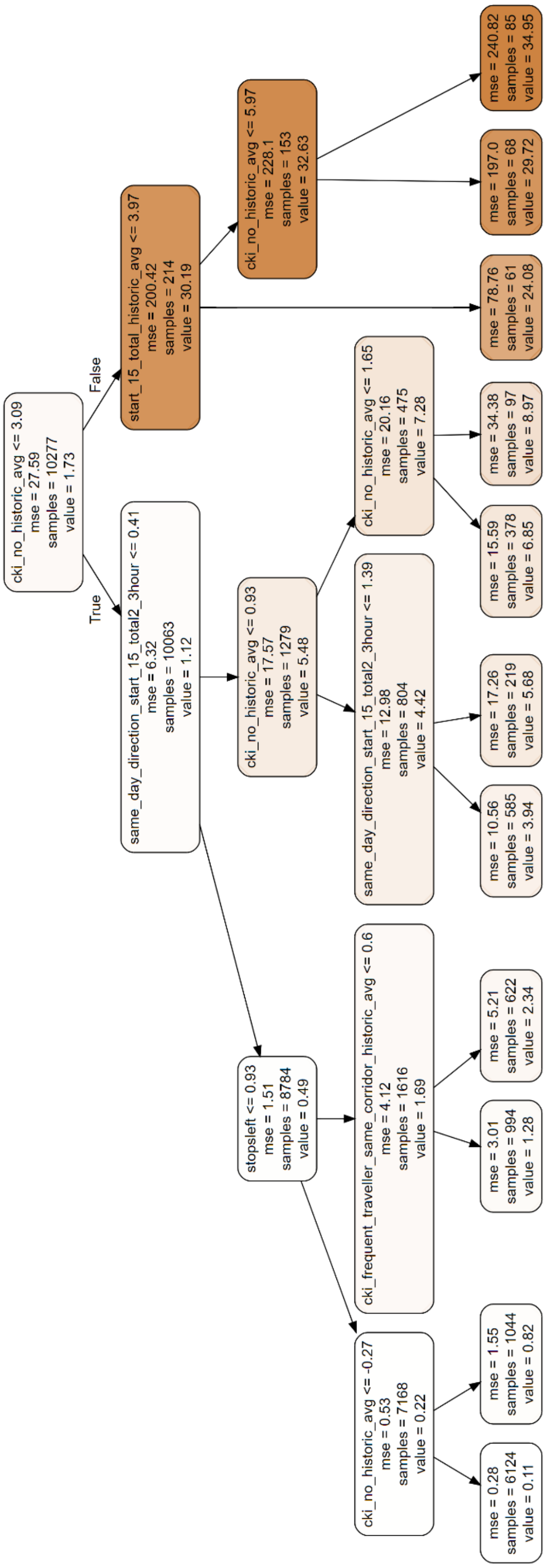


Figure S.1: An example of a trained decision tree model

Appendix T History trip planners

The way of informing the travelers about the bus schedules has changed. In the beginning of the 19th century these schedules were published in the local newspapers. (Since the schedules of buses repeated each day or week, you only needed to present one typical day or week per line.) Later, as the bus operators fused into larger companies with more lines, the bus schedules were collected and put in a booklet (Dutch: busboekje). The booklet is still operational with some design changes. However, because of the ease of use, most people use online interactive information platforms, like the trip planner of 9292, to plan their journey using public transport. These online travel planners are still improving. Some improvements are usability oriented, but others change the functionality as well. For instance, since 2014 9292 makes use of real-time travel information (Redactie OV-Magazine, 2014).

Appendix U Trip characteristics in the Netherlands

This appendix discusses the trip characteristics statistics as reported by CBS.

Figure U.1 shows the average number of trips per person per day in the Netherlands. Beside the total, the number of trips is also shown per mode. It should be noted that a multimodal trip is only counted once and thus attributed only to the dominant mode. Nevertheless, from Figure U.2 we conclude that the private modes are more popular.

Figure U.3 shows that when people use the bus/light rail/metro as dominant mode, over half of the trips are generated for nonrecreational activities, such as commuting to work, school or university.

Figure U.4 shows that the average trip distance by bus is around 12 km. This is similar to the 11 km average trip distance as published by OV-bureau for the 2 million travelers in February 2017 (http://ovbureau.nl/ov-cijfers/dashboard/). The trip purposes commuting to work, school and universities and social visits have a slightly higher average of between the 14 and 18 km, whereas the purposes shopping and services/healthcare are around 7 and 8 km.

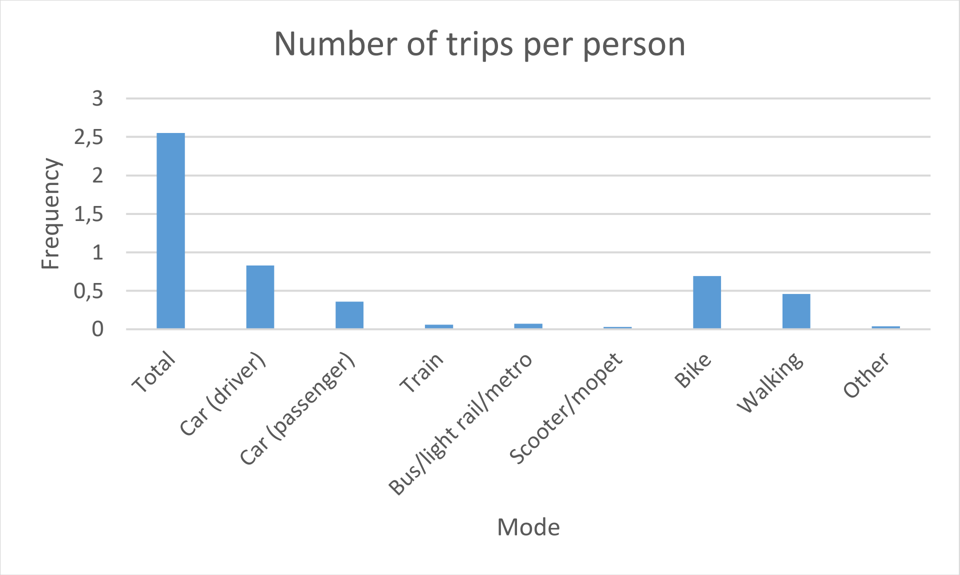


Figure U.1: The average number of trips per person per day per mode where multimodal trips are only counted once and attributed to the dominant mode (Central Bureau for Statistics, 2017b). The data are for 2016 in the Netherlands.

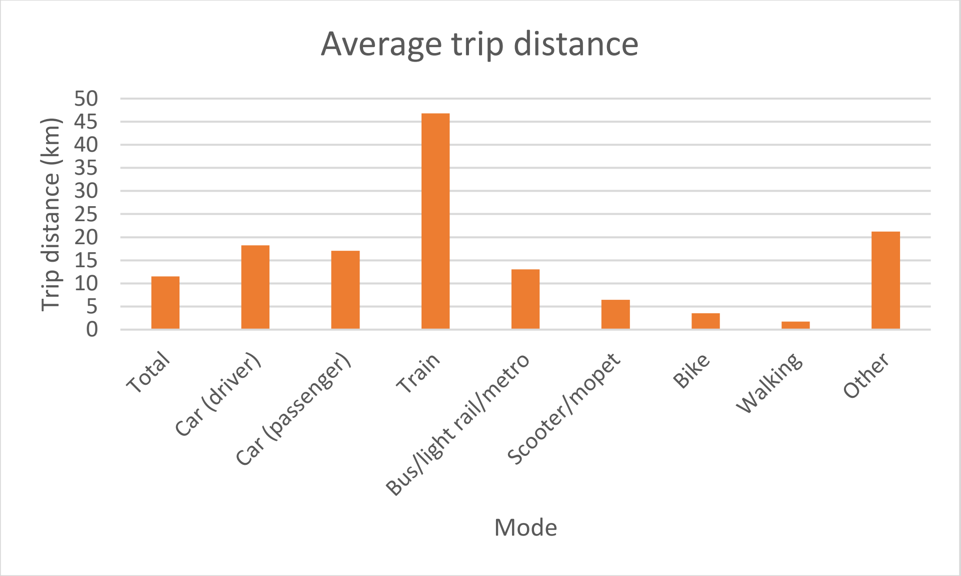


Figure U.2: The average trip distance per mode in 2016 in the Netherlands (Central Bureau for Statistics, 2017b). Each leg in a multimodal trip is taken into acount for the relevant mode.

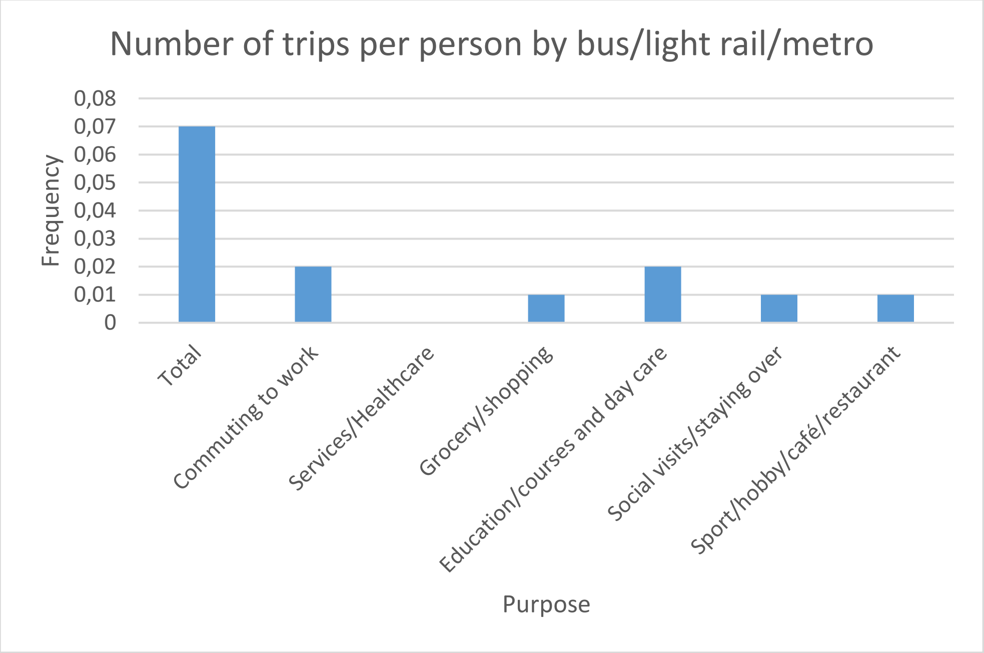


Figure U.3: The average number of trips per person per day per trip purpose by bus/light rail/metro in the Netherlands for 2016 (Central Bureau for Statistics, 2017b). If this mode was not the dominant mode in a multimodal trip, the trip is not counted. There was no data for the trip purposes business and touring/hiking so these are neglected.

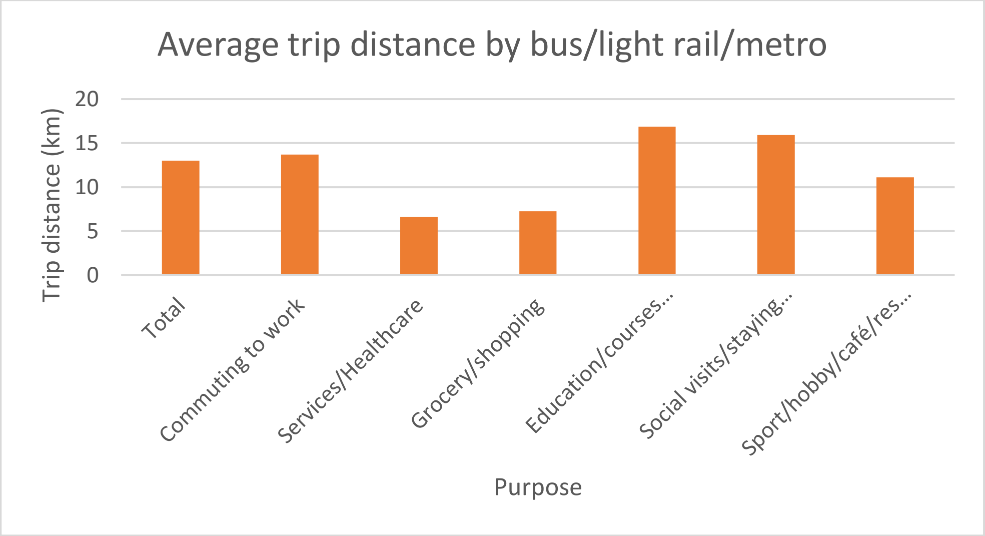


Figure U.4: The average trip distance per trip purpose by bus/light rail/metro in the Netherlands for 2016 (Central Bureau for Statistics, 2017b). The distances for the use of this mode, dominant or not, for multimodal trips are included. There was no data for the trip purposes business and touring/hiking, so these are neglected.

Appendix V Feature importance

In this appendix the figures of the feature importance per data partition are shown. The feature importance is abstracted from the Random Forest *Regressor* as implemented by Scikit-Learn. The importance of a feature will vary per model type, number of features included and data partition. Therefore, these scores only give an indication. Beneath we present the feature importance for the boarding and alighting model for the data partitions *g554-1-0/workday* and *g554-1-0/Workday 8 AM.*

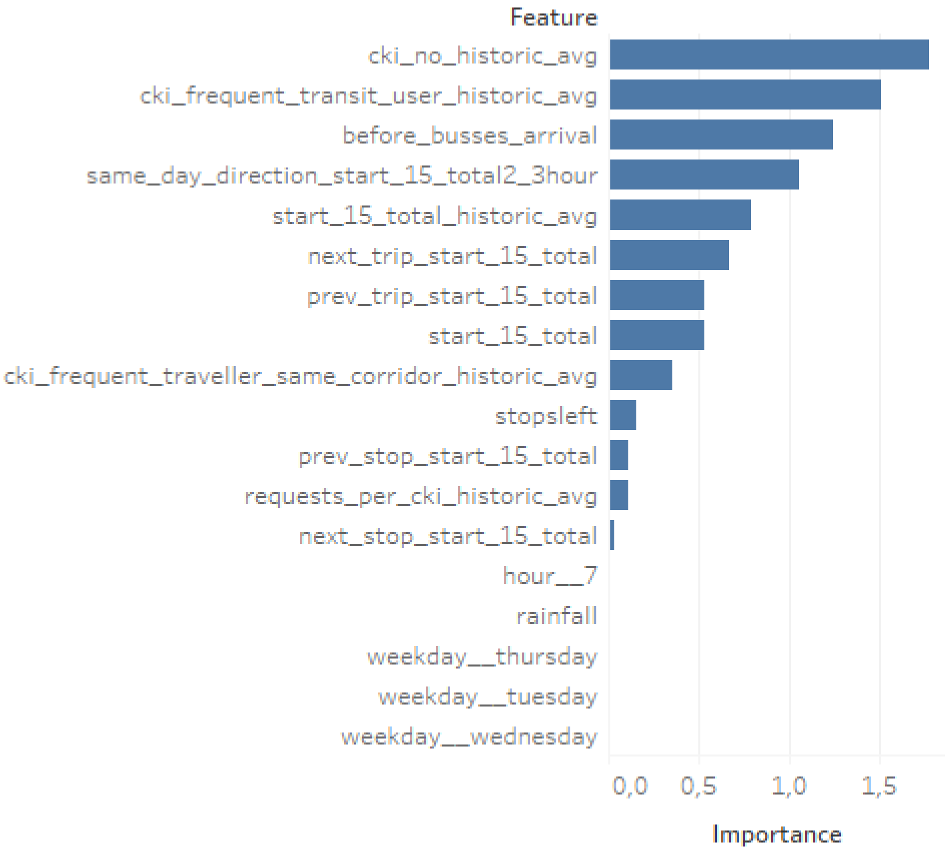


Figure V.1: Feature importance of the boarding model for the line variant g554-1-0 and during the morning peak (Workday 8 AM)

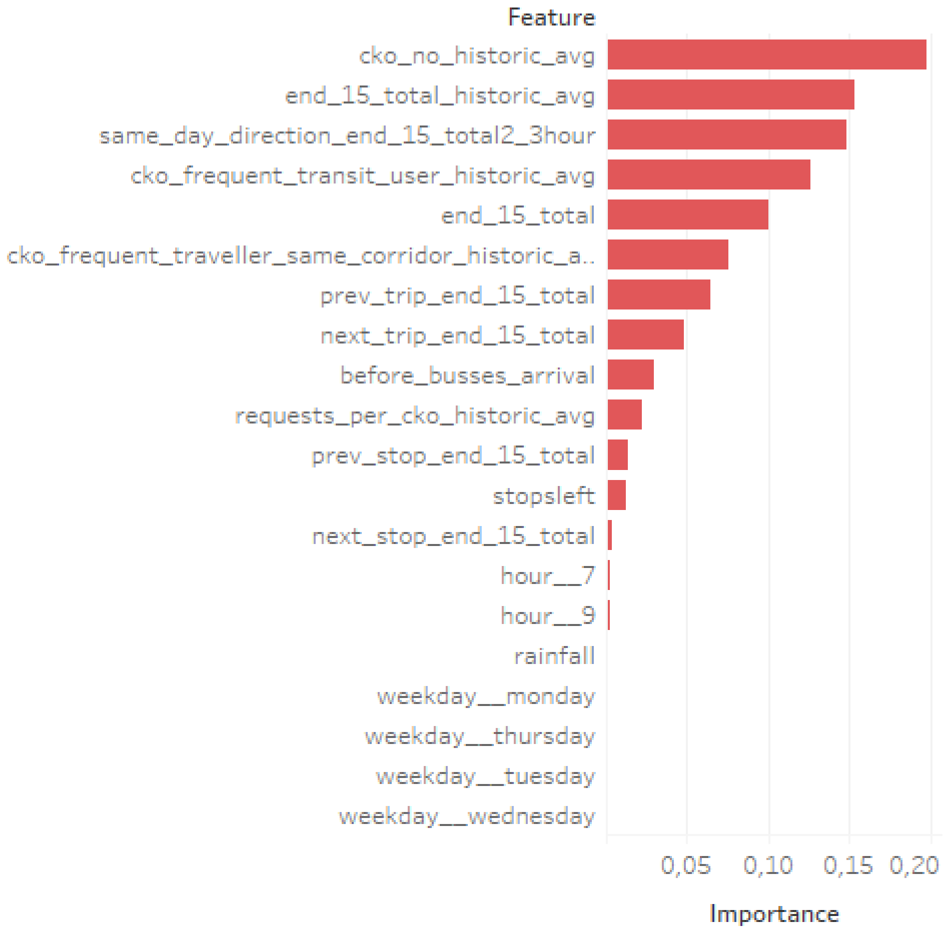


Figure V.2: Feature importance of the alighting model for the line variant g554-1-0 and during the morning peak (Workday 8 AM)

Appendix W RMSE per stop cluster

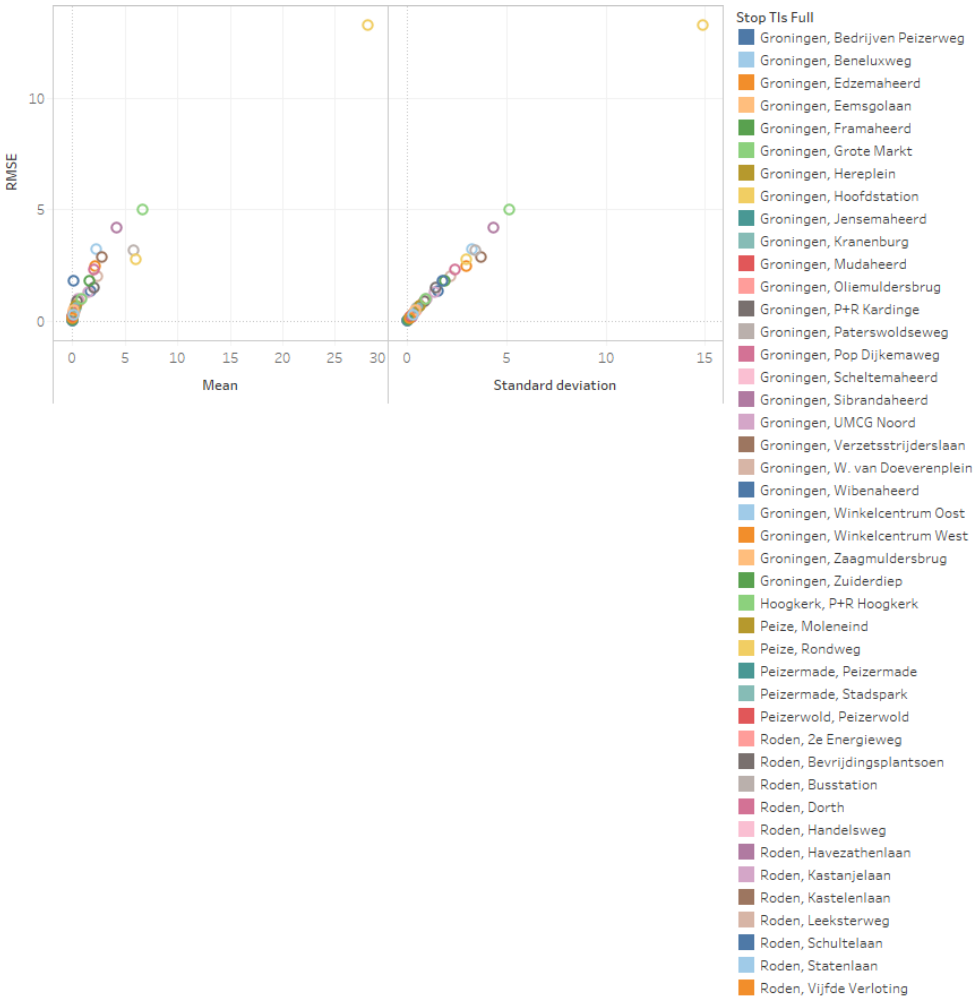


Figure W.1: The RMSE per stop cluster plotted versus the mean and standard deviation of people boarding the stop cluster. Used model: Boarding, RF, g554-1-0, Workday

Appendix X Effect of subsampling

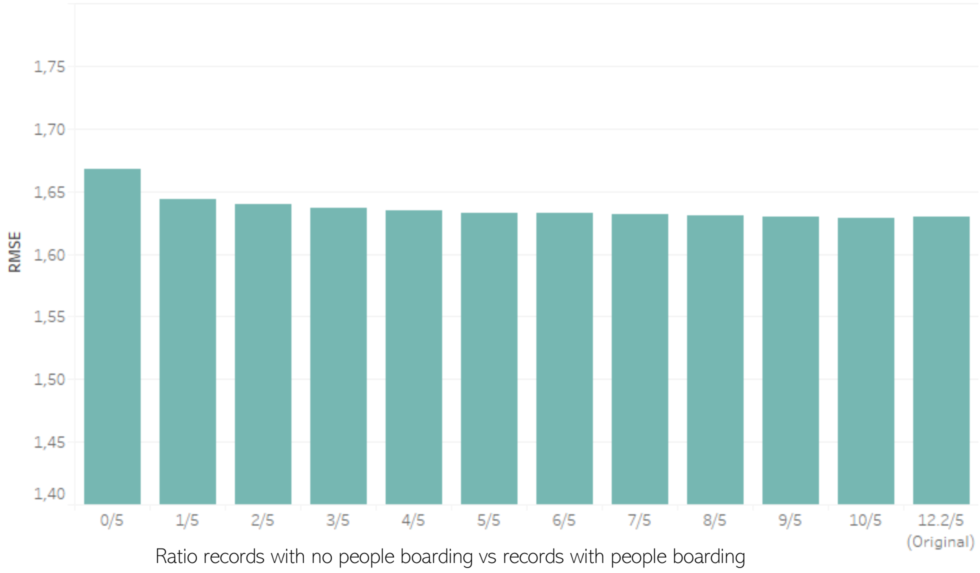


Figure X.1: The effect of subsampling on the RMSE for the boarding model. Using the best performing model (RF, 20 features, workday/g554-1-0 partition). Y-axis starts at 1.4.

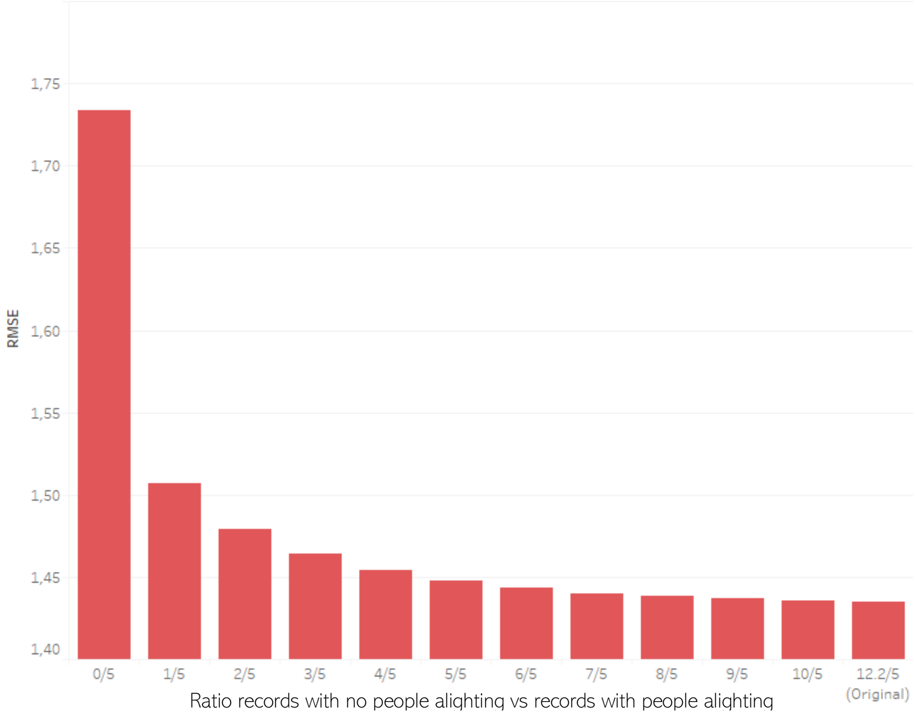


Figure X.2: The effect of subsampling on the RMSE for the alighting model. Using the best performing model (RF, 20 features, workday/g554-1-0 partition). Y-axis starts at 1.4.

Appendix Y Bus types

Qbuzz operates different kinds of vehicles, see Table Y.1. In the near future they will also implement a 15 meter double decker for the Qliner and 12 meter H^2 bus and an 18 m articulated electric bus. The different buses have different seat and crush capacities, which should be taken into account when using the forecast for bus allocation (Van Oort, 2015a). It could be that the different type of buses causes different habits and thus demand characteristics. However, data on which bus is used is missing, therefore we cannot take this into account.

Network level	12 m	15 m	18 m Articulated bus[2]	20 m Articulated bus	22 m articulated bus
City	✓		✓		
Regional	✓		✓		
Qliner	✓	✓			
Qlink				✓	✓

Table Y.1: The different kind of buses OV-bureau and Qbuzz utilize within the different network levels.

[2] Dutch: Gelede bus or Harmonicabus