Real-Time Passenger Load Estimation using In-Vehicle Data

Thesis Report

D. van Gelder

Delft University of Technology

TU Delft

Image credit: HTM Media Center

Real-Time Passenger Load Estimation using In-Vehicle Data

Thesis Report

by

D. van Gelder

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday 27th June, 2022.

In collaboration with Siemens Mobility

SIEMENS

Student number:	4551028	
Project duration:	September 1, 2021 – June 27, 2022	
Thesis committee:	Dr. Neil Yorke-Smith,	TU Delft, Responsible Supervisor
	Dr. Anna Lukina,	TU Delft, Daily Supervisor
	Dr. Oded Cats,	TU Delft, Co-Supervisor
	Dr. Guohao Lan,	TU Delft
	Drs. Emilio Tuinenburg,	Siemens Mobility

This thesis is confidential and cannot be made public until June 27, 2024.

TU Delft

Abstract

Increased urbanisation has led to significant challenges for public transport operators. Inconsistent demand leads to peaks in passenger activity on the network. Moreover, the COVID-19 pandemic has introduced a need for social distancing as well, limiting the desired capacity of vehicles. To combat this, intelligent real-time and data-driven decision making is required. In many cases, the data required is lacking or not available in real-time. Our research addresses these challenges by providing means to gain insight into the passenger load of public transport vehicles. The focus of this research is to investigate how using in-vehicle sensor data can help in constructing an estimate of the passenger load and evaluate its contribution.

By combining in-vehicle sensor signals with historical passenger flow patterns, a novel fusion model based on gradient boosting machines is constructed that can make real-time predictions of the passenger load using this data as input features. The evaluation shows that its estimates have a mean absolute error (MAE) score of 7.83, outperforming a random forest model baseline by 37%. Moreover, a crowding indicator analysis demonstrated that when predicting crowding indicators, the model achieves a weighted F1 score of 0.828. An ablation study found that excluding the in-vehicle features from the model reduces the model's performance significantly, it could reduce the performance by up to 42%. In fact, the same experiment showed that having only the in-vehicle features is preferable to historical passenger flow features. Therefore, we conclude that using in-vehicle sensor data can be a feasible alternative to historical AFC data for predicting the passenger load.

The methodology has been extended by constructing a short-term forecasting model based on Seasonal ARIMA and GARCH that uses real-time signals of the passenger load to update its forecasts. The results show that while the forecasts lack accuracy initially, once the model is updated the forecasts improve up to almost a negligible error. When model predictions are used as update signals, the forecasting model is still able to improve and the results are competitive, despite the error contained in the signals.

Overall, we conclude that the proposed real-time model offers a suitable method for passenger load prediction and clearly demonstrates the effectiveness of using in-vehicle sensor data as input features. Moreover, we have presented a feasible method for using these features in a forecasting setting with a real-time model as an intermediary.

Preface

Over the past academic year, I have been working on this thesis, the product of which is the report that you are now reading. While I was originally given quite a practical topic, it ended up providing lots of opportunities for theoretical analysis and additional research avenues. I have found it very interesting to study this topic and unravel all the insights from my experimental results. While most of the work has been done from home, the Siemens Mobility office has provided me with a warm environment to discuss any challenges I faced as well as get suggestions to improve my work.

Just as I complete this thesis, so too do I conclude my six years of studies at the TU Delft. This has been a very formative part of my life that resulted in unforgettable experiences and close friendships. I have been incredibly privileged to be able to pursue my studies with the freedom and joy that I have had, for which I am grateful.

I would not have been able to complete this thesis without the help of a small army of people. Their help and support is deeply appreciated and I would like to acknowledge the support of a few persons in particular.

First of all, I would like to extend my deepest gratitude to my TU Delft supervisors Anna Lukina, Neil Yorke-Smith and Oded Cats for their guidance, as well as their thoughtful suggestions and feedback throughout the entire thesis process. Of course, my thanks also go to my colleagues in the Digital Lab at Siemens Mobility. In particular, I would like to thank my supervisors Emilio Tuinenburg and Danny Meringa, without whom this thesis would not have been possible and whose supervision has helped me stay on the right *track* (pun intended). Finally, yet just as importantly, my gratitude also goes to my dear friends and family who have always provided me with relentless and unwavering support, especially when I was at my wits' end.

I sincerely hope that the reader finds this report as insightful and enjoyable as I have intended it to be. One important lesson I have received early on in my studies is that a technical presentation is no good without a poetic afterthought. Therefore, I will conclude with the following words:

"Do or do not, there is no try."
– *Yoda, 4 ABY*

D. van Gelder
Rotterdam, May 2022

Contents

1 Introduction

This thesis addresses the problem of passenger load prediction. In tackling such a problem, a wide variety of related topics are explored. Therefore, in this chapter, we will provide an introduction to the main research themes and describe the relevance of the problem in a broad context. The research objectives are identified in terms of research questions and the consequent contribution that this work aims to make is outlined as well. Finally, we provide an overview of the structure of the report.

1.1. Problem Context

Public transport is a critical part of the infrastructure of countries across the globe. Especially in urban environments, it plays a crucial role in reducing congestion and greenhouse gas emissions. The International Energy Agency (IEA) has estimated that 40% of the mobility sector's greenhouse emissions are produced by urban transport: over 4 billion tonnes of CO_2 annually as of the year 2021 [1]. One of the most effective methods to reduce these emissions is by encouraging commuters to use public transport more often [2]. In addition, with an expected increase in the urbanisation of the population, the demand for public transport will increase as well.

For public transport operators, this poses a significant challenge. While often servicing a large urban area, peak moments of demand require ad hoc management such that the load on the transport network can be adequately alleviated. Moreover, operators wish to optimally utilise their rolling stock to ensure that the vehicle is rarely empty but also never completely full or overloaded. This is beneficial to both the comfort of the traveller and may also reduce the wear on the vehicle's parts.

A recent additional challenge is posed by the COVID-19 pandemic. which has been shown to have had a significant effect on public transport ridership [3]. It also introduced a need for social distancing between passengers which puts additional constraints on the desired passenger load in the vehicle.

To address these challenges, intelligent decision making is required. However, to do so, insight into the state of the public transport network is necessary with information about the passenger load in individual vehicles. To facilitate ad hoc decision making, this information should ideally be available in real-time. Many operators rely on real-time Automated Fare Collection (AFC) or Automatic Passenger Counting (APC) data to acquire this information. Due to a variety of reasons, this information may not be available (in real-time). For instance, installing APC sensors into vehicles can be expensive or privacy regulations prohibit operators to process AFC data directly.

A solution could be provided by the vehicle itself. Modern public transport vehicles, such as trams, collect and store a large number of sensor data for maintenance and diagnostic purposes. This data could also be used to derive insight into the current passenger load. By combining this sensor data with knowledge about historical passenger load patterns, an estimation of the passenger load can be made. While this would provide only an approximation of the actual passenger load, it may help the transport operators in addressing the aforementioned challenges by providing real-time passenger load information.

Figure 1.1: Image of an Avenio tram as manufactured by Siemens Mobility. Image retrieved from the Siemens Mobility Website.

.

1.2. Research Question

This thesis aims to tackle the problem of real-time passenger load estimation, based on in-vehicle data complemented by historical passenger load data. The selected transport context is tram vehicles. More specifically, trams manufactured by Siemens Mobility[1] and operated by a tram operator in the Rotterdam – The Hague metropolitan area. This operator will be referred to as "the current tram operator" throughout this work. An example of the tram vehicle under consideration, the Avenio model, is shown in Figure 1.1. Due to the novelty of using such in-vehicle features, besides investigating the effectiveness of a model that estimates the real-time passenger load, we consider how well the in-vehicle data contributes to an accurate estimation. Moreover, investigating how incoming real-time vehicle data may improve short-term passenger load forecasts is part of the research as well.

The main research question that is thus investigated is:

> How accurately can a model based on real-time tram vehicle sensor data and historical passenger flow create a real-time estimate of the passenger load in a tram vehicle?

To further concretise the research objectives, we provide the following sub-questions to formulate the research goals in more detail:

- RQ1: What features of the available real-time vehicle data are most descriptive of or correlated with the passenger load?
- RQ2: What is the best method, in terms of estimation accuracy, to make real-time passenger load estimations?
- RQ3: What effect does excluding the historical passenger load data have on the accuracy of the estimating model?
- RQ4: How can the real-time data be incorporated into a new model to make short-term passenger load predictions for the remainder of a vehicle trip?
- RQ5: What is the best method, in terms of prediction accuracy, to make short-term passenger load predictions using real-time vehicle data and passenger load estimations as well as historical passenger flow records?

[1] https://www.mobility.siemens.com/

1.3. Contributions

This work aims to add to the growing number of research works that introduce novel approaches to the passenger load prediction problem. This research in particular provides a novel approach, using a combination of multiple data sources which have, as of yet, not been considered in conjunction with each other in the literature. Within the current context, the proposed solution aims to provide a better understanding of passenger load patterns throughout time. Furthermore, it seeks to provide insight into the contribution of in-vehicle data as a complement to historical passenger load data, or as standalone information. This could demonstrate that intelligent estimation methods could provide passenger load information using pre-existing data sources that would otherwise be provided by costly measurement equipment.

While the research is set in a specific context, namely tram vehicles in the Rotterdam – The Hague metropolitan area, this work could be applied to different transport modes with similar data sources or entirely different urban contexts. In fact, the same tram vehicles that are operated by the current tram operator are operational in many different cities around the globe.

1.4. Report Organisation

The organisation of this thesis report is as follows. Chapter 2 provides an overview of related work as well as scientific background on the important methods used in this research. The methodology and evaluation criteria of the research are introduced in Chapter 3. A description of the process of data engineering and the construction of the dataset is provided in Chapter 4. Chapter 5 contains an analysis of the features with respect to the target variables of interest in this work using the constructed dataset. Both real-time and forecasting models are designed and selected in Chapter 6. The experimental evaluation results and analyses of these results are provided in Chapter 7. Finally, a conclusion to the research as well as a discussion of the findings is provided in Chapter 8.

2 Background

In this chapter, we provide the reader with relevant background material regarding the topics covered in this work such that they have an understanding of those topics as well as the context of the current research with respect to the state-of-the-art literature. Where applicable, useful references to other literature are made to acquire a more in-depth discussion of certain topics. The first section covers related literature on passenger load and flow prediction as well as forecasting. Then, gradient boosting machines are introduced, a type of model that this work relies upon. In addition, an overview of some model interpretability techniques is provided. Finally, we provide a description of the time-series forecasting methods applied in this work.

2.1. Related Work

In the following sections, a structured overview of the related work with regard to both the real-time prediction and the forecasting of the passenger load and flow is described.

2.1.1. Passenger Load and Flow Prediction

In this section, we discuss the related literature on passenger load and flow prediction. Here, there is a distinction between the passenger *load*, the absolute number of passengers on board, and the passenger *flow*, how individuals and groups of passengers travel in a public transport system. However, the literature may also use these terms interchangeably, partially depending on the perspective of prediction that is employed. It is important to note that if an accurate passenger flow estimate has been achieved, the passenger load can be estimated as well, but not vice versa.

The organising principle behind the discussion of the literature is to find out what input data sources are used in the study as well as the generic method utilised to solve the problem. The results of this investigation are summarised in Table 2.1.

One of the indicators of passenger load is the dwell time of a vehicle at a stop. Often, methods use Automatic Vehicle Location (AVL) combined with General Transit Feed System (GTFS) data to estimate the current load of vehicles [4, 5]. Moreira-Matias and Cats [4] construct a passenger load profile for a bus trip using constrained local regression. A recent study by Sun et al. [5] constructs a probabilistic model for passenger flow values and solves using Bayesian inference. Zhang et al. [6] combine Automatic Fare Collection (AFC) data of smart-card users with an estimation of the number of coin-paying passengers based on bus dwell times to create an overall load estimate that can be used for forecasting. They found that the number of coin-paying passenger boarding times follows a Poisson distribution. This prior information is used to create a load estimate for this group of passengers

If AVL data is complemented with historical Automatic Passenger Count (APC) or real-time AVL data, more accurate estimates can be derived as well. For instance, Jenelius [7] has combined GTFS and real-time AVL data with historical APC data to provide Real-Time Crowding Information (RTCI) to passengers. RTCI provides additional information to normal load estimation, as it also takes into account the capacity of the vehicle. Various regression techniques (LASSO regularised regression and multivariate PLS regression) are effectively applied. However, it is noted that using real-time APC data will significantly improve the results.

Study	AVL	APC	GTFS	Other	Solving Method
[4]	✓		✓		Constrained Local Regression
[5]	✓				Probabilistic Modelling
[6]	✓			Real-time AFC	Probabilistic Modelling
[7]	✓	✓	✓		Various Regression Techniques
[9]				Weight	N/A
[11]	✓		✓	Vehicle Dynamics	Arithmetical
[10]				Weight, Date	Linear Regression

Table 2.1: Comparison of types of input data used for the real-time passenger load estimation methods in the literature.

Combining traffic data sources such as GTFS, AFC/APC and AVL data is a non-trivial task. A methodology to combine such data sources is provided by Luo et al. [8], which is especially relevant for this research as the data in that work was provided by the current tram operator.

Limited research has been directed towards load estimation using other vehicle-related data sources such as engine power or weight. Jenelius [9] provides RTCI based on real-time load data derived from metro vehicle weight, showing more accurate results than using historical averages. In this case, the method used to estimate the passenger load based on vehicle weight is not provided. Nielsen et al. [10] construct an estimate using linear regression, taking into account various random effects besides the vehicle weight, such as the day of the week, season and metro line number. They demonstrate that these methods may be more effective than traditional APC measurement methods like infrared sensors. A study by Kovacs et al. [11] uses information about a tram car's driving mechanics and information about the track to construct a load estimate. A limited case study indicates accurate results. To the best of the author's knowledge, no other work has been done that combines a set of multiple data sources originating from the vehicle to create a load estimate.

Another interesting research direction for passenger load estimation is based on using WiFi [12, 13] and/or Bluetooth [14, 15] signals within a vehicle. These approaches demonstrate accurate results as well but are out of scope for this research.

2.1.2. Passenger Load and Flow Forecasting

Similar to the previous section, the investigation of the relevant literature for passenger load and flow forecasting is guided by an organising principle, albeit slightly broader. In this case, the guiding principle is to investigate what techniques were employed in the relevant literature and the specific setting of the forecasting method. Factors such as the perspective, station-centric or vehicle-centric, or the number of steps ahead have a big influence on the performance of the method. Hence, it is important to take into account what methods are applied under what circumstances. As the literature on forecasting is more extensive, it allows for a more in-depth exploration of the works compared to the previous literature overview. The results of this investigation are shown in Table 2.2.

Passenger load and flow forecasting is "a time-series, non-linear, random and unstable problem, which depends mainly on copious amounts of high-quality data and methodologies" [16]. It has received considerable research attention where approaches often focus on either short-term forecasting, up to one hour ahead in time, or long-term forecasting, up to years ahead in time [17]. This research often occurs in a more data-rich scenario, where GTFS, AVL and APC/AFC records are available in real-time.

Techniques for short-term traffic forecasting can be divided into two categories [6, 18]: parametric and non-parametric techniques. One of the most successfully applied parametric methods is autoregressive integrated moving average (ARIMA). It was first applied to traffic forecasting in the 1970s [19] but, due to its success, ARIMA is still applied in passenger flow forecasting, often as part of a multi-stage pipeline [20–24].

ARIMA is a powerful method for forecasting the expected mean of a time series. However, it may sometimes fail in modelling volatility in the time series. Ding et al. [20] and Chen et al. [25] address this by modelling the volatility using GARCH and complementing the estimate by ARIMA with a variance term. However, the results vary depending on the degree of volatility that is apparent in the time series. For instance, in an earlier study by Chen et al. [26] in predicting traffic flow, it was found that under regular circumstances of flow there is less significant volatility which reduces the effectiveness of the

GARCH model. Hence, it seems that GARCH may contribute in situations where there is significant unexpected volatility. A few examples would be public events or national holidays.

A successfully applied (non-parametric) method in passenger flow prediction is Kalman filtering[1] [28]. It is often combined with other methods to fine-tune a prediction. For instance, Wang et al. [18] demonstrate a two-stage bus passenger load prediction method where the first stage is an adaptive Kalman filter method on a stop level and the second stage uses Support Vector Regression (SVR) to predict at the bus level.

Similarly, a recent approach by Li et al. [23] combines Seasonal ARIMA and SVR to predict short-term passenger flow, taking into account both real-time passenger flow and external factors like the weather.

Gong et al. [21] propose a three-stage passenger flow prediction method. Specifically, they aim to predict the waiting passenger count at bus stops as well as the number of boarding and alighting passengers. The stages combine the ARIMA method, with an event-based algorithm and finally a Kalman filtering method in order to make this estimation.

The previously mentioned study by Zhang et al. [6] also applies Kalman filtering when predicting busloads. It is used to calibrate the prediction derived from similar historical passenger flow patterns.

Recent research has also attempted to apply Deep Learning (DL) methods to the problem of passenger flow prediction. Due to the time-series nature of passenger flow patterns, a commonly applied DL technique is Long Short-Term Memory (LSTM) neural networks [29]. Due to their ability to capture both long- and short-term relationships in time series data, they provide a suitable candidate for passenger flow prediction by nature [17, 30, 31].

Toqué et al. [17] use an LSTM model for short-term forecasting. However, a competitive Random Forest (RF) model seems to outperform the LSTM model. The approach by Pasini et al. [31] uses an LSTM model with an encoder-decoder architecture. This model is thought to learn representations better which can be exploited for passenger flow prediction. The results indicate that this architecture outperforms the traditional LSTM and indeed learns representations in the data. This indicates that incorporating domain knowledge into a model may help in achieving better results.

Liyanage et al. [32] applied a BiLSTM model to passenger load forecasting from a vehicle perspective, significantly outperforming a comparable regular LSTM model. The results showed that the performance is highest for the shortest forecasting steps, 15 minutes in advance, which degrades over longer horizons. In addition, they note the potential improvement that can be made by incorporating additional features such as the weather.

Another interesting result is put forward by Guo et al. [30] who combine LSTM with SVR. The authors argue that SVR is better at capturing stable patterns from historic samples and LSTM is more capable of predicting abnormal fluctuations. The results demonstrate that a combined model of both LSTM and SVR is more effective than separate models.

Liu et al. [33] propose a modular LSTM architecture where separate LSTM networks are trained based on different urban rail network features (e.g., cyclical, spatial-temporal, etc.) meant to incorporate domain knowledge. The outputs of these modules are combined through a fully connected network. The use of domain knowledge seems to improve the network's performance.

Finally, Yang et al. [34] argue that the performance of LSTM can be improved with more extensive feature engineering. By intelligently combining historical data (Enhanced Long-Term Features) with real-time data, a better prediction is possible.

Ensemble methods such as gradient boosting machines and random forest models have been successfully applied in forecasting as well. A recent work by Shiao et al. [35] applies a random forest model to passenger flow forecasting using varying sets of features. The results indicate that the applied model is optimal with the largest feature set, containing historical passenger flow information. This insight is confirmed by Liu et al. [36] who found that a combination of temporal information with historical passenger flow information are the optimal features for a random forest model.

A recent work by Gallo et al. [37] has applied gradient boosting machines to short-term passenger load forecasting, both from a vehicle perspective as well as a station perspective. They combine a wide variety of features such as the weather and transfers with other stations. The results indicate that gradient boosting machines are suitable for both scenarios of passenger load prediction. The authors note, however, that performance degrades over larger forecasting horizons. Ding et al. [38] predict short-term subway ridership using various gradient boosting machines using both historical subway

[1] An introduction to Kalman filtering can be found in the work by Bishop and Welch [27].

ridership values as well as bus transfer activities and temporal characteristics. Interestingly, using a split-count relative importance measure, the authors were able to identify the relative importance of features. The analysis showed that the latest subway ridership value has a relative importance of over 80% in the applied model. As the previous study indicates, ensemble methods can be used for variable selection. Zhao et al. [39] propose a feature selection method using the ensemble models' variable importance measures. Again, the latest passenger flow data appears to be the most important feature. Moreover, brief model analysis is performed and the ensemble models both outperform both various ARIMA and LSTM models, with a gradient boosting model being the optimal model.

2.1.3. Discussion and Research Gaps

While the literature on passenger load/flow prediction is limited, the research indicates that most methods rely on either (linear) regression or probabilistic modelling to derive their estimation. The input data is mostly real-time AVL and GTFS data while other in-vehicle data such as engine power, historical AFC/APC, etc., are mostly not considered.

The literature on passenger load/flow prediction/forecasting is more extensive. However, it seems that most studies adopt a station-centric approach which is inherently different from the current context. Interestingly, classical algorithms like ARIMA are still widely used, although often in conjunction with machine learning techniques. Another interesting observation from the literature is that incorporating domain knowledge, like the cyclical nature of passenger flow patterns, has a clear benefit on prediction accuracy. The most predominant non-parametric techniques are Kalman Filtering, SVR and LSTM neural networks. In some cases, these techniques are combined in multi-stage pipelines in order to exploit each method's strengths. Interestingly, often only a limited amount of input features is considered and methods mostly rely on incoming signals of the passenger load or flow. Ensemble methods such as random forests and gradient boosting machines have been successfully applied as well. It is not clear which specific method is optimal and it ostensibly depends largely on the problem scenario and modelling. Remarkably, only a limited set of features is considered in most research. Most of the methods consider temporal features and average or real-time passenger flow features. However, external factors such as the weather, dwell time or lateness of the vehicle seem to be rarely considered.

Overall, there is little comparative research between state-of-the-art methods. In some cases, a comparison to a baseline method like ARIMA is made (for instance, see the study by Liu et al. [33]). The reason for this is most likely two-fold: first, the source code of almost all methods is not publicly available, making it difficult to replicate. Second, the literature uses a wide variety of datasets that are proprietary, no public benchmarking dataset seems to exist. Moreover, most research in forecasting is performed from a station-centric perspective as can be seen in Table 2.2, which is an inherently different scenario than a vehicle-centric approach. Another notable gap is a discussion of the forecasting horizon in the passenger flow forecasting literature. Most often, a predefined forecasting horizon is evaluated. This could be several steps ahead of the current time step [18]. In these cases, a forecast could be fine-tuned using incoming real-time information of the passenger flow. However, this is not considered in the current literature.

Based on the literature overview in both passenger load estimation and passenger flow prediction, the following gaps in the academic literature have been identified: (1) The limited research that addresses real-time passenger load focuses mostly on a combination of real-time AVL and historical APC data . (2) No research has been performed as to which features are relevant for an accurate passenger load estimate. (3) Little comparative research between state-of-the-art methods has been performed. (4) The forecasting literature is often limited to a station-centric approach. (5) No investigation of how real-time signals can improve the forecast for longer horizons while forecasting the passenger load or flow has been conducted.

Study	Target	Proposed Method(s)	Perspective	Mode(s)	Interval	Features	Steps Ahead	Baseline(s)
[20]	Load	ARIMA + GARCH	Station	Subway	15 minutes	Real-time load	Unknown	
[25]	Flow	ARIMA + GARCH	Station	Subway	10 minutes	Real-time flow	Unknown	
[18]	Load & Flow	KF + SVR	Station & Vehicle	Bus	15 minutes	Average and real-time flow, average load, headway	1-3 steps	LR, LASSO
[23]	Load	ARIMA + SVR	Station	Subway	15 minutes	Average and real-time flow	Unknown	ARIMA, SVR
[21]	Flow	ARIMA + KF	Station	Bus	5 minutes	Average and real-time flow	Unknown	
[6]	Load	KF	Vehicle	Bus	Every Stop	Average and real-time flow	2 steps	ARIMA, LR
[17]	Flow	RF, LSTM	Station	Train, Tram, Bus	15 minutes	Day of week, public holiday, average flow	1 step	LR
[31]	Load	LSTM	Vehicle	Train	2-182 minutes	Day of year, day type, minute, train route, delay, past 6 values of load and flow	1-6 steps	GBM, LSTM, LR
[30]	Flow	SVR + LSTM	Station	Subway	15 minutes	Day of week, time of day, public holiday, recent flow	1-2 steps	NN+RF+SVR+KNN, ARIMA, SVR, LSTM
[33]	Flow	LSTM + NN	Station	Subway	10 minutes	Day of week, weather precipitation, station distances (travel time), real-time and 4 past values of flow	Unknown	KNN, ARIMA, NN
[34]	Flow	LSTM	Station	Subway	1 hour	Average and real-time flow	17-85 steps	LSTM, ARIMA, SVR, NN
[32]	Load	LSTM	Vehicle	Bus	15, 30, 60 minutes	Real-time flow	1 step	LSTM, NN (various)
[35]	Flow	RF	Station	Subway	1 hour	Last 5 values of flow, day of week, public holiday	1 step	
[36]	Flow	RF	Station	Subway	1 hour	Year, month, day of month, week and hour, public holiday, lunar month, lunar day, previous average flow, previous 10 values of flow	1 step	
[38]	Flow	GBM	Station	Subway	15 minutes	Time of day, day of month, day of week, last 4 values of transferring flow, last 3 values of passenger flow	1 step	
[39]	Flow	GBM, RF	Station	Subway	15 minutes	Stations (clustered), rain, average flow, day of week	1 step	LSTM, RF, ARIMA, NN
[37]	Load	GBM	Station & Vehicle	Tram	Every Stop	Last 8 values of load, real-time flow, departure times, weather, connections, overlappings with other lines	1-8 steps	

Table 2.2: Comparative overview of the passenger load and flow forecasting literature describing: the target variable (load or flow), the proposed methods, the forecasting perspective (station or vehicle), the transport mode under consideration, the data interval, the features used, the number of steps ahead in prediction and the baselines used for comparison. The methods are abbreviated as follows: Linear Regression as LR, LASSO Linear Regression as LASSO, (Seasonal) Auto-Regressive Integrated Moving Average as ARIMA, Generalised Auto-Regressive Conditional Heteroskedasticity as GARCH, Kalman Filtering as KF, Support Vector Regression as SVR, Random Forest Regression as RF, Long Short-Term Memory as LSTM, (Feed-Forward) Neural Network as NN, Gradient Boosting Machine as GBM and k-Nearest Neighbors as KNN. A plus sign indicates that a combination of methods is proposed rather than a set separate methods.

2.2. Gradient Boosting Machines

Gradient Boosting Machines have achieved much attention as a general-purpose machine learning method. Easy-to-use open source libraries such as XGBoost [40] and scikit-learn [41] have contributed to the popularity of this method. As this machine learning method is relied on in this work, we will briefly introduce the intuition behind this method as well as its mathematical definition.

The idea of gradient boosting machines was first introduced by Friedman [42]. It can be considered as a form of gradient descent combined with boosting. The intuition behind gradient boosting is that by sequentially fitting weak learners that improve on earlier weak learners, one ends up with a strong ensemble learner. Whereas some ensemble methods such as random forests fit their weak learners independently and produce predictions by means of averaging [43], gradient boosting is an iterative method that builds and predicts sequentially. While it can be used for both regression and classification, we will restrict ourselves to the regression case as it is more relevant in the context of the current work.

A gradient boosting machine consists of three components: an additive model $F_M(x)$, a set of M weak learners $h_1(x), h_2(x), ..., h_m(x)$ and a loss function $l(y_i, F_M(x_i))$. The additive model whose prediction is $\hat{y}_i$ for input x_i is of the following form (in a regression case):

$$\hat{y}_i = F_M(x_i) = \sum_{m=1}^{M} h_m(x_i) \tag{2.1}$$

As a general rule, the weak learners used are Classification and Regression Trees (CART) as introduced by Breiman et al. [44]. The additive model ensemble is constructed greedily through its recursive definition:

$$F_m(x) = F_{m-1} + h_m(x) \tag{2.2}$$

The weak learner $h_m(x)$ is constructed to minimise the sum of losses over the input set by the previous additive model F_{m-1} as follows:

$$h_m = \arg\min_h \sum_{i=1}^{N} l(y_i, F_{m-1}(x_i) + h(x_i)) \tag{2.3}$$

By means of first-order Taylor approximation, the loss for an instance i is approximated as follows:

$$l(y_i, F_{m-1}(x_i) + h_m(x_i)) \approx l(y_i, F_{m-1}(x_i)) + h_m(x_i) \left[\frac{\delta l(y_i, F(x_i))}{\delta F(x_i)} \right]_{F=F_{m-1}} \tag{2.4}$$

where the term $\left[\frac{\delta l(y_i, F(x_i))}{\delta F(x_i)} \right]_{F=F_{m-1}}$ denotes the derivative of the loss with respect to $F(x_i)$, which is F_{m-1} at evaluation. Assuming that the loss is differentiable, its computation is straightforward. This differentiation is denoted by g_i. As such, the newly constructed tree h_m can be formulated as follows:

$$h_m \approx \arg\min_h \sum_{i=1}^{N} h(x_i) g_i + c \tag{2.5}$$

where c denotes constant terms from the previous equations. A minimised $h(x_i)$ therefore predicts the negative gradient of the previous ensemble: F_{m-1}, through steepest descent. Hence, the procedure can be considered a form of gradient descent. Note that the loss function is not restricted to squared errors, Friedman [42] describes several loss criteria: least-squares, least absolute deviation, Huber loss and logistic binomial log-likelihood.

Some improvements to the gradient boosting machine's original definition have been made by introducing regularisation. Two main methods for regularisation exist that are most commonly applied: shrinkage and subsampling [45].

The first introduces a shrinkage parameter in the form of a "learning rate" parameter v to the recursive step in Equation (2.2) yielding:

$$F_m(x) = F_{m-1} + v \cdot h_m(x) \tag{2.6}$$

It has been demonstrated that smaller values of v lead to a better generalisation error but increase the training risk [42]. Observe from Equation (2.6) that the values of the learning rate v and the number of trees M are closely related and should thus be optimised in conjunction.

The second regularisation technique, subsampling, aims to improve the accuracy by exposing the individual trees to a subsample of the training data. In practice, it reduces the computation time of the algorithm and seems to improve the accuracy, especially when combined with shrinkage [45, 46].

2.3. Model Interpretability

The previous section has introduced the concept of gradient boosting machines. This method, as well as other ensemble and DL methods, can be considered black-box ML models. While the logic behind the methods may be understood, it is still a highly complex or impossible task to decompose the model's output. Hence, we rely on analytical methods that assist us to get an understanding of the logic behind a model's decision. In the context of this work it is important to understand the impact of each individual input feature on the model's output, as posed by research question RQ1. Hence, the model's decisions with respect to the input features need to be evaluated.

In this section, we will discuss several relevant methods for model interpretation, especially with respect to gradient boosting machines. For a discussion regarding the motivation for interpretability in black-box machine learning models as well as alternative approaches to making models interpretable, we refer to Lipton [47] and Guidotti et al. [48].

The analysis of a model's interpretability results in so-called *feature importances* where the relative impact of a feature on the output is quantified for each feature. These can be constructed on a global level as well as on an individualised level. Global feature importance measures for tree-based methods fall into three categories [49]:

- **Gain**: a feature's importance is based on the total reduction of loss or impurity through each split depending on that feature. This is a classic approach introduced by Breiman et al. in 1984 [44], but still in wide use.
- **Split**: this method counts the number of splits depending on a feature and takes it as an importance measure, the intuition being that a feature that splits the data often is relatively important [38].
- **Permutation**: the permutation importance methods rely on permuting the values of a feature in the test set and observing the change of error by the model compared to the non-permuted test set. If the error is larger, this indicates that the feature's importance is relatively larger as well. See Auret et al. [50] for a further discussion of the properties of permutation feature importances.

These approaches have recently been applied in the passenger load prediction literature as well [39, 51]. For a comprehensive overview of other global feature importance measures, also for non-tree-based models, we refer to Wei et al. [52].

A crucial shortcoming of these methods is that they provide feature importances on a global level, aggregated over the whole dataset. It could be useful to understand the influence of features on the instance or individualised level. Depending on the scenario of an instance, feature importances may vary in a practical setting.

The literature on individualised model interpretability, especially for tree-based models, is less established. Nonetheless, several methods exist, varying in their explanatory power. The most simple method is to use Individual Conditional Expectation (ICE) plots [53]. It is quite similar to partial dependence plots [45] but on an individualised level. It visualises the dependence of the prediction on a specific feature for varying feature values. As it is an individualised method, it can do so for each instance individually. However, it is able to do so meaningfully only for one feature at a time. It also may produce infeasible plots if some features are correlated. Individualised methods are also referred to as local methods.

A variant of ICE plots is counterfactual explanations, which is a description of the smallest change to the feature values that changes the prediction to a predefined output [54].

A popular local method is called Local Interpretable Model-agnostic Explanations (LIME) [55]. The main interpretability method applied here is the construction of local surrogate models that explain the model predictions in a certain region of the input space. LIME generates a perturbed dataset

and the corresponding predictions of the “black-box” model. The new samples are weighted based on their proximity to the instance of interest. An interpretable model, e.g. LASSO or CART, is then fitted to this weighted dataset such that it is accurate locally. As a result, these local surrogates do not have an accurate global approximation. The degree to which the surrogate is locally accurate is referred to as *local fidelity*. The complexity of the surrogate model is defined beforehand. That is, the number of features that the surrogate model fits to. In fact, a surrogate model may fit to different features than the original model. This may be interesting for text classification, where word embeddings could be extracted for the surrogate model. The challenge in this method lies in defining a suitable neighbourhood, which is a complex task. Moreover, if a poorly defined neighbourhood is used the results might be negatively affected.

A recent and alternative method for interpretability is called SHapley Additive exPlanations (SHAP) [56]. It is a game-theoretic approach to model interpretation, by applying the concept of Shapley values [57] to feature values. The intuition is that the prediction task for a single instance can be modelled as a game, where the feature values are the players that form coalitions to receive a payout, which is the predicted value. Each feature value contributes some amount towards the total payout. Hence, the marginal contribution of a feature value is its Shapley value. This assumes that the payout is a sum of the marginal contributions, thus that each feature contributes *additively* to the prediction which is a strong assumption. As the computational complexity increases exponentially in the number of features, calculating Shapley values directly is infeasible.

The SHAP method proposed by Lundberg and Lee [56] contains approximation methods in order to make the calculation of Shapley values. One of the approximations is referred to as KernelSHAP, it applies LIME with certain parameters such that it yields consistent Shapley values.

The most relevant approximation with respect to the current work is the method for approximating Shapley values for tree-based methods, called TreeSHAP [49]. It estimates the SHAP values in polynomial time: $O(TLD^2)$ where T is the number of trees, L the maximum number of leaves in each tree and D is the tree depth. This makes the method particularly useful for interpreting gradient boosting machines as the weak learners are typically trees with low depth. Moreover, the SHAP method is consistent which is a property that other methods often lack. This means that if a model prediction relies more on a feature value than another model for a particular instance, that the Shapley value is always higher for that model’s feature value [56]. Finally, the method allows the extraction of global feature importances based on the local explanations by aggregating them. Calculating the overall feature importance for a feature is simply done by taking the mean of the sum of the absolute Shapley values for all instances.

Overall, the SHAP interpretability method provides a mechanism to collect insights into feature effects both on a global and a local level. The global feature importances can be used to rank features based on their importance but local SHAP values can be used to quantify the specific effects of each respective feature value on the output of the model for a specific instance.

2.4. Time Series Forecasting

In this section, the methods of time series forecasting that this work relies on are elaborated upon. As is evident from Section 2.1.2, time series forecasting methods such as ARIMA are popular in the state-of-the-art literature regarding passenger load/flow prediction. The purpose of this section is thus to inform the reader of the mathematical background of these methods and to provide an intuition about why these methods are suitable in the context of the current work.

2.4.1. Seasonal ARIMA

The Auto-Regressive Integrated Moving Average (ARIMA) model is a method for forecasting time series. It is a generalisation of the Auto-Regressive Moving Average (ARMA) model. While the ARMA model is restricted to forecasting for stationary time series, the ARIMA model is suitable for forecasting non-stationary time series by introducing a method for differencing observations by another [58]. The ARIMA model consists of three components: Auto-Regressive (AR) component, the Integration component (I) and the Moving Average (MA) component [59].

An AR model considers the variable of interest to be regressed on its prior values. Hence, it models the variable as a linear combination of a certain amount of previous values. The order, denoted by p, determines the amount of lagged values to model. An AR model of the order p can thus be formulated

as follows:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \epsilon_t \tag{2.7}$$

where c denotes the intercept, ϕ_t denotes the regression coefficient for a lagged value at time step t and ϵ_t models white noise.

An MA model considers the regression error to be a linear combination of error terms whose values occurred at the current and various times in the past. An output value y_t can thus be thought of as a weighted moving average of the past few forecast errors. An MA model in the order q is formulated as:

$$y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q} \tag{2.8}$$

where, again, c denotes the intercept, θ_t denotes the regression coefficient for a forecast error at time t and ϵ_t is white noise.

The final integration component, in the order d, refers to the times first-degree differencing is applied to the data. Here, integrated refers to the reverse of differencing. The operator for differencing is denoted as ∇. Second order differencing ($d = 2$) can therefore be formulated as [60]:

$$\nabla y_t = y_t - y_{t-1}$$
$$\nabla(\nabla y_t) = \nabla y_t - \nabla y_{t-1}$$

A non-seasonal ARIMA model is thus defined by the three parameters (p, d, q). To simplify the notation, the lag operator L is often used to denote lagged values. A general formulation of the model is as follows:

$$(1 - \phi_1 L - \ldots - \phi_p L^p)\, \nabla^d y_t = c + (1 + \theta_1 L + \ldots + \theta_q L^q)\epsilon_t \tag{2.9}$$

Forecasting is then performed by simplifying the equation such that y_t is on the left-hand side and new points are forecasted iteratively for each step on the forecasting horizon. Future observations are replaced by their forecasts, future errors by zero and past errors with the corresponding residuals [58].

Many time series exhibit regular patterns on a higher level: this is referred to as seasonality. This is a characteristic of a time series where the data experiences regular and predictable changes in a certain time period, typically one year. When an ARIMA model also includes parameters for modelling seasonality, the model is refered to as Seasonal ARIMA, often abbreviated to SARIMA. A Seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA model which are applied over all seasons. The parameters are then referred to as (p, d, q) for the non-seasonal part of the model and $(P, D, Q)_m$ for the seasonal part of the model where P, D, Q refer to the AR, I and M seasonal components and m to the periodicity of the season, respectively. The periodicity is the number of observations per season and needs to be constant over the time series. The mathematical formulation of the model is extended as follows:

$$\begin{aligned}(1 - \phi_1 L - \ldots - \phi_p L^p)(1 - \tilde{\Phi}_1 L^m - \tilde{\Phi}_P L^{P \cdot m})\nabla^d \nabla_m^D y_t = A(t) \\ + (1 + \theta_1 L + \ldots + \theta_q L_q)(1 + \tilde{\Theta}_1 L^m + \ldots + \tilde{\Theta}_Q L^{Q \cdot m})\epsilon_t\end{aligned} \tag{2.10}$$

which can be abbreviated to:

$$\phi_p(L)\tilde{\Phi}_P(L^m)\nabla^d \nabla_m^D y_t = A(t) + \theta_q(L)\tilde{\Theta}_Q(L^m)\epsilon_t \tag{2.11}$$

where $\phi_p(L)$ is the non-seasonal AR lag polynomial, $\tilde{\Phi}_P(L^m)$ is the seasonal AR lag polynomial, $\nabla^d \nabla_m^D y_t$ is the differenced time series, $A(t)$ is the trend polynomial which is often only a constant intercept c, $\theta_q(L)$ is the non-seasonal MA lag polynomial and $\tilde{\Theta}_Q(L^m)$ is the seasonal MA lag polynomial.

The first step in building a forecasting model is to find the order parameters (p, d, q) $(P, D, Q)_m$. A structured method for acquiring these (Seasonal) ARIMA as well as ARMA parameters has been proposed by Box and Jenkins [61], referred to as the Box-Jenkins method. It requires manual inspections of autocorrelation function (ACF) and partial autocorrelation function (PACF) plots. Automated methods rely on criteria such as the Akaike Information Criterion to find the optimal parameters [62].

Once the model order has been found, the regression parameters ϕ_p, θ_q and possibly $\tilde{\Phi}_P$ and $\tilde{\Theta}_Q$ need to be estimated. Many implementations estimate it through maximum likelihood estimation (MLE) [58], which is similar to least squares estimation for the regression equation in Equation (2.11) [59].

2.4.2. Exogenous variables

The formal definition of ARIMA in Equation (2.11) includes a "trend polynomial", which is in essence an intercept. One could replace this intercept by linear regression on explanatory variables. The model is then transformed to a form of linear regression with ARIMA errors. These explanatory variables are referred to as exogenous variables and are the independent variables with respect to the dependent variable (i.e., endogenous variable) [63]. Other works refer to exogenous variables as covariates [64]. The exogenous variables need to adhere to some properties for them to be useful for the model. First, they need to be linearly related to the endogenous variable. Second, they need to be stationary in the same order as the endogenous variable. Finally, the values variables need to be known over the entire forecasting horizon.

If these conditions are met, then the exogenous variables can be added into the regression model as follows [63]:

$$\begin{aligned} y_t &= \beta_t x_t + u_t, \\ \phi_p(L)\tilde{\Phi}_P(L^m)\nabla^d\nabla_m^D u_t &= A(t) + \theta_q(L)\tilde{\Theta}_Q(L^m)\epsilon_t \end{aligned} \tag{2.12}$$

where β_t is the vector of regression coefficients for the exogenous variables and x_t is the vector of the exogenous variable values. Observe how the Seasonal ARIMA equation is essentially modelling the error.

2.4.3. GARCH

While (Seasonal) ARIMA models can be considered to model the conditional mean of a stochastic process over time. Auto-Regressive Conditional Heteroskedasticity (ARCH) has been proposed to model the conditional variance of that process. ARCH has been proposed by Engle in 1982 [65] and is appropriate when a time series error variance follows an AR model. Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) has been proposed as a generalization, where the process may also follow an MA model [66]. Hence, it is quite similar to an ARMA model.

A GARCH model for a process is defined by two parameters: (p, q). Given a process of a stochastic error with variance σ_t^2 and an information set ψ_t such that $\epsilon_t|\psi_t \sim N(0, \sigma_t^2)$, the GARCH model is defined as:

$$\begin{aligned} \sigma_t^2 &= \alpha_0 + (\alpha_1\epsilon_{t-1}^2 + \ldots + \alpha_p\epsilon_{t-p}^2) + (\delta_1\sigma_{t-1}^2 + \ldots + \delta_q\sigma_{t-q}^2) \\ &= \alpha_0 + A_p(L)\epsilon_t^2 + D_q(L)\sigma_t^2 \end{aligned} \tag{2.13}$$

where α_i and δ_i refer to the regression coefficients for the error and variance which are summarized to $A_p(L)$ and $D_q(L)$, respectively[2]. Note the similarity of the model to the AR and MA models defined in equations (2.7) and (2.8). The GARCH(1,1) model is typically used as it is the most simple and most robust configuration [67].

(Seasonal) ARIMA can be complemented by GARCH by combining the two methods into a single equation for modelling the process of interest. In this case, ARIMA models the conditional mean and GARCH models the conditional variance of the process. Essentially, GARCH would be fitted to the residuals of the ARIMA model. This approach has also been applied in the context of the current work [20, 26]. Consider a process of variable y_t to be modelled by (Seasonal) ARIMA and GARCH as follows:

$$y_t = u_t + \epsilon_t, \qquad \epsilon_t \sim N(0, \sigma_t^2) \tag{2.14}$$

then u_t is defined similarly to u_t in Equation (2.12) and σ_t^2 as σ_t^2 in Equation (2.13). As an additional variance to the ARIMA prediction, it can model volatility in the time-series whereas ARIMA assumes a variance that is homogeneous. These two models can thus complement each other in such settings.

[2] The original paper uses the notation β_i for the MA part of the equation. Here, δ_i is used to avoid confusion with the regression coefficients for exogenous variables in Equation (2.12).

3

Methodology

In Section 1.2 we have outlined the main research questions of this thesis. In this chapter, we will translate these questions into concrete research objectives and an approach to fulfilling these objectives. In addition, we provide an introduction to the evaluation metrics that are used to evaluate the proposed models. We introduce these in this chapter, as these criteria are referred to from different chapters throughout the work.

3.1. Overview

Before describing each of the steps in the methodology of this work, we will provide a brief overview and rationale of the methodology.

A crucial element in the analysis and evaluation of the current research is the dataset. This dataset needs to consist of the various feature and target variables under consideration. Hence, a data engineering pipeline is constructed that combines the available data sources and extracts and aggregates relevant information. Afterwards, an analysis of the relationships between the features and the passenger load and passenger flow is necessary. This is done partly to answer research question RQ1, but also to derive insights into the problem's nature to incorporate into the model design. To derive these insights, both a quantitative and qualitative analysis is performed on the relationships.

Using these insights, a model can be constructed that is able to make real-time passenger load estimations. The designed model is evaluated on a test dataset in a defined experimental setup. In addition, the results are compared to a baseline which is comparable with the literature. This evaluation addresses research question RQ2.

To further evaluate the model and the contribution of the vehicle features, an ablation study is performed where the set of vehicle features is eliminated from the model as well as the set of historical features. This is done to quantify the contribution that the vehicle data may provide to a model, as formulated in research question RQ3.

Besides the real-time estimation model, a forecasting model is designed and evaluated as well. This model makes short-term forecasts of the passenger load on the trip level. It should incorporate information from real-time information to make more accurate forecasts. The design of the model addresses research question RQ4 and the evaluation of the model addresses research question RQ5. In the following sections, the methodology is described in more detail.

3.2. Data Engineering

As outlined in the previous section, the initial task of the methodology is to model the high-frequency time-series sensor data into a format that fits into a model. It is not feasible or even desired to make estimations for each separate vehicle sensor signal. The estimation to be made is concerning the passenger load at *some* point in between two stops. This would require some aggregation of the data. Moreover, multiple data sources need to be integrated: real-time sensor data from the Avenio tram vehicles, tram timetables (GTFS) and historical passenger flow data (AFC). The methodology of Luo et al. [8] may form a theoretical basis for engineering this integration. The goal is to create a

data engineering pipeline that first preprocesses the applicable data sources, then integrates them and finally extracts the features from the data in order to create a dataset.

3.3. Feature Analysis

We have described that a dataset is to be constructed through a data engineering pipeline. This dataset will contain a variety of features which may originate from the vehicle, AFC or GTFS data sources. To understand how each of these features may contribute to a passenger load estimate, an empirical feature analysis is performed. Performing statistical analyses may be useful for investigating a relationship between the input features and the actual passenger load. This may also be performed by analysing the produced model by looking at which features contribute to the model's output. Besides analysing the relationship to the passenger load, the relationship of the feature to the passenger flow is of importance as well. These are two related aspects of the state of the vehicle and thus relationships to either of these may be taken into consideration for the model design.

The quantitative assessment will form the basis of the feature analysis by using empirical methods to find possible relations between the features and the target variables. This will consist of applying various statistical measures to discover both linear and non-linear relationships between the features and the passenger load/flow. Three measures are applied to the relationship between each feature and both the passenger load and flow. The first measure is the Pearson correlation coefficient. It is the covariance of the two variables divided by the product of their standard deviations, mathematically defined as follows:

$$\rho_{X,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{3.1}$$

where X and Y denote the two random variables under consideration.

The second measure that is applied is univariate linear regression which tests the effect using the feature as a single regressor. It will convert the correlation coefficient defined in Equation (3.1) to an F statistic. The associated p value is obtained by applying the F-test survival function.

The final measure is Mutual Information (MI), which measures the mutual dependence between two variables X and Y [68]. It quantifies the *information gain* obtained about one of the variables after observing the other variable. In probabilistic terms, it compares the difference of the joint distribution of X and Y, referred to as $P_{X,Y}$, and the product of the two marginal distributions P_X and P_Y. It is defined as follows:

$$I(X;Y) = D_{\mathrm{KL}}(P_{X,Y} || P_X \otimes P_Y) \tag{3.2}$$

where D_{KL} is the Kullback–Leibler divergence. The reason for including this measure is that it is not limited to continuous variables or linear dependencies but can quantify any statistical dependency. It is therefore useful for comparing discrete features to the passenger load/flow as well as capturing non-linear relationships.

With the results of the quantitative analysis, a qualitative analysis of the features is performed by means of visual inspection of the relationship between the features of particular interest and the passenger load or flow. This may provide insight into the specific type of relationship that the feature exhibits. Specific types of relationships are explored based on the domain knowledge regarding the feature. The visual inspections may indicate interesting aspects of relationships as well as the non-existence of relationships, despite the results of the quantitative analysis. Moreover, they may indicate that the relationship is stronger for some specific values of the feature.

After both evaluations, a discussion of the results will be provided where the identified relationships are examined. Finally, for each feature, there will be a discussion about its relevance and potential as a predictor for the passenger load/flow.

3.4. Model Design

By incorporating the insights from the feature analysis described in the previous section, a model can be constructed to estimate or forecast the passenger load. As posed by research questions RQ2 and RQ4, models need to be constructed that make real-time predictions and forecasts of the passenger load. These will be separate models as the problem nature varies depending on the prediction scenario. More concretely, there are no real-time sensor signals available in a forecasting setting. Therefore, a different model is most likely more suitable for that scenario. The methodology for selecting and designing the models is described in the next sections.

3.4.1. Real-Time Model

While the literature commonly uses either probabilistic modelling or linear regression to perform passenger load estimation [4–6, 10], our current problem is in a much larger feature space, in terms of the number of features. These models aren't able to capture the possible non-linear relationships in the data, thus more complex models need to be considered. Possible candidates would be support vector machines, ensemble models and gradient boosting machines. Work in passenger flow forecasting has shown that ensemble methods, as well as support vector machines, have been successfully applied.

It may be useful to use some of the insights from the feature analysis in determining the exact model. Moreover, the structure of the problem as being a time series could be exploited as well. In the forecasting literature, many approaches use a multi-stage pipeline with a fusion of multiple model predictions [18, 30, 36]. This could be relevant for the real-time model as well and such model designs are considered as well. Ultimately, the design of the real-time prediction model should assist in answering research question RQ2.

The selection of the specific model will rely on a brief analysis of a small train and test set to evaluate which shows the most promising results. Then, the best model is selected and further optimised towards the current context. This could be decomposing the model into several models making specific predictions for either the passenger load of flow as well as finding the optimal parameters through a grid search of possible parameters.

3.4.2. Forecasting Model

The related work overview in Section 2.1 has shown that a wide variety of techniques has been applied for passenger load forecasting. The model design for the current forecasting model should select a competitive model that has successfully been applied in the literature and expand on it to address research questions RQ4 and RQ5.

The goal of the forecasting model design is to create a model that can incorporate real-time signals to derive a better forecast. In addition, historical data can be incorporated as well to improve the forecasts. Based on the literature, it is evident that forecasting methods predominantly use one of four methods: ARIMA, SVR, Kalman Filtering or LSTM neural networks. Hence, a selection should be made out of these four models.

To address the research gaps outlined in Section 2.1.3, the model should be applied in a vehicle-centric context where passenger load forecasts are made from the perspective of the vehicle over the remainder of the trip rather than the station. In addition, another research gap can be addressed by incorporating real-time signals and updating the model to evaluate the effect of these signals on the forecasts.

3.5. Ablation Study

Another key objective of the research, outlined in research question RQ3, is to investigate whether either excluding historical passenger load data or the real-time vehicle data as a set of input features has a negative effect on estimation accuracy. This may also be referred to as an ablation study specifically aimed at removing these data sources.

In practice, it may be desirable to have a solution that solely relies on in-vehicle data, as (historical) AFC data may not be available in all contexts. Moreover, a comparative study of which set of features produces a more accurate model may also demonstrate the effectiveness of having this in-vehicle data, which is a part of the novelty of this work. The same experiment as mentioned in Section 3.6.1 will be considered, evaluating the same model design but without each set of features. As multiple models are proposed, the best performing model is selected to perform the ablation study on based on the evaluation of the real-time model. A comparative analysis of the results should illustrate the effect of excluding either component and whether the accuracy is affected significantly.

3.6. Evaluation

The evaluation is a critical part of the research, as it assesses the performance of the implemented models described in Section 3.4. The procedure and metrics of this evaluation are referred to in several places in this report. Therefore, the evaluation is described in this section for both the real-time models and the forecasting models. As separate models are defined for the tasks of real-time predictions and forecasting, the evaluation will also evaluate these tasks independently.

3.6.1. Real-Time Model Evaluation

A suitable method to evaluate the performance of the developed model, besides using error measures such as the RMSE, MAE, etc., is to compare it with other models from the literature. However, these models do not have open source code available so they need to be reproduced. Simple baselines can be implemented such as models purely based on historical averages. As an example, see the calendar model proposed by Toqué et al. [17]. An experiment is set up, incorporating a comprehensive sample of the dataset to use for fitting and evaluating the models. The accuracy of the proposed model may be determined through an error measure like the root-mean-square error (RMSE). However, a more thorough error analysis is required as well, investigating in what circumstances the model performs better and worse.

Three evaluation metrics are used to indicate the performance of the models on the test set, namely the root-mean-square error, the mean-absolute error and the coefficient of determination. Each metric highlights a different aspect of performance and thus having multiple metrics may contribute to achieving a better understanding of the models' performance. In the description of each of the metrics, the following terminology is used:

- $\hat{y}_i$, the predicted output value for test set member i
- y_i, the actual output value for test set member i
- $\overline{y}$, the mean actual output value
- N, the size of the test set

The Root-Mean-Square Error (RMSE) measures the mean of squared errors and provides an overall sense of goodness of fit. However, it penalises larger errors as all errors are squared. Due to its sensibility to outliers, it gives a good indication of how well the model generalises over all the samples. A lower RMSE score indicates a better goodness of fit, with 0 being the optimal score. The RMSE is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{3.3}$$

The Mean Absolute Error (MAE) is a similar measure to the RMSE. It measures the average error in absolute terms. Each error contributes to the overall score in proportion to the absolute value. It is a more interpretable measure, as it gives an indication of the expected deviation of the model. Note that due to its non-sensitivity to outliers, its value for a model can be significantly different than the RMSE score. Two models with the same MAE value, may have different RMSE scores. It thus allows models to be compared over different dimension. In this case, a lower score indicates better goodness of fit where, again, a value of 0 is the optimal score. It is defined as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{3.4}$$

The coefficient of determination (R^2) represents the proportion of variance of the output that has been explained by the independent variables in the model. A higher score indicates better goodness of fit. A model that outputs a constant value will receive a score of 0 but the score can also be negative if a model is worse than the constant model. It has an upper bound of 1. As it is a measure of explained variance, it can indicate how well a model captures the distribution of the predicted data. It is defined as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(\overline{y} - y_i)^2} \tag{3.5}$$

Besides evaluating the prediction performance of the passenger load value, the real-time models are also evaluated on their ability to predict crowding indicators [6, 7, 9, 37] in addition to regression estimates. In other words, that is predicting a certain level of crowding in the vehicle. This is a common phenomenon in public transport planners. An example of how crowding indicators are communicated

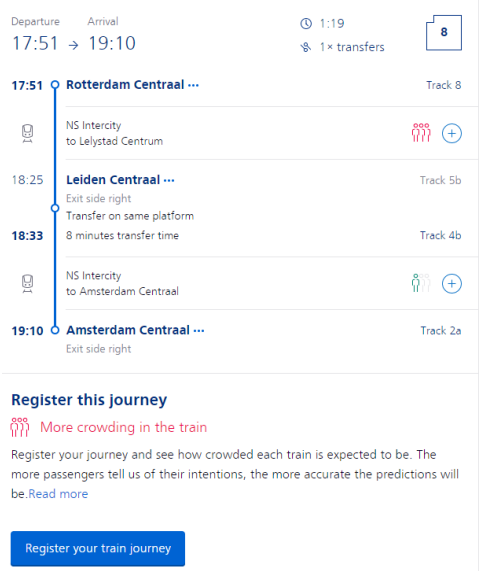

Figure 3.1: Image of the travel planner from the Dutch Railway service (NS) showing various crowding indicators on a planned journey to the right of the image.

to passengers is shown in Figure 3.1. In this case, the passenger load values are grouped into bins, which each indicate a degree of crowding.

As the crowding indicator evaluation transforms to problem from a regression to a multi-class classification problem, the previously proposed metrics are no longer suitable. Therefore, a common metric used for classification is used: the F1 score. It is mathematically defined as follows:

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{3.6}$$

where TP represents the number of true positives, FP is the number of false positives and FN is the number of false negatives.

The F1 score is the harmonic mean of the precision and recall. It can be calculated for each target class. In order to arrive at a single value for the overall performance, two methods are used. The macro F1 score describes the mean of the F1 score of each respective class. On the other hand, the weighted F1 score weighs the F1 score of each class by its support, i.e. the number of target samples in that class. The weight F1 gives a proportional score while the macro F1 provides emphasises the performance on classes with a low support more. Both methods are computed for the results.

3.6.2. Model Interpretability

To better understand the predictions of the designed real-time model as well as the effects of individual features on the output, a method of model interpretability is applied. The goal is to identify how feature values affect the predictions of the model and whether the feature importances in the model align with the strength of features identified in the feature analysis.

Given that the real-time model is a black-box model, a specialised method needs to be applied for model interpretation. For this, the SHAP method is used as described in Section 2.3. The main advantage of using SHAP is that it has a specialised polynomial approximation algorithm for tree-based models as well as its ability to compute both local model explanations and global feature importances.

The global feature importances can be used to compare the feature importances of the real-time model to the relationships found in the feature analysis.

Local explanations are used to understand how individual feature values have an impact on the model. Due to the size of the dataset, an exhaustive analysis cannot be made. But it may be possible to understand the type of influence that input features may have on the prediction by considering aggregated local explanations. These are aggregated results of individual local explanations visualised such that the relation between the feature and the output under varying circumstances can be clear.

3.6.3. Forecasting Model Evaluation

When evaluating the forecasting model it is crucial to make an overall comparison to a different model as well as evaluate the effect of incoming real-time signals. For a proper evaluation of the forecasting model, a real-time scenario should be simulated which may occur in practice as well, where the model can update based on incoming data.

To the best of the author's knowledge, no evaluation metrics for iteratively updated forecasting models exist. Therefore, handcrafted metrics are used to evaluate the effect of incoming observations. The goal is to quantify the effect of these incoming observations as well as how the forecasting horizon influences the performance. In the discussion of the handcrafted metrics, the following terminology is used:

- m, the length of the trip, which is constant for all test set trips
- n, the number of trips in the test set
- $\hat{y}_{i,j,k}$, the forecast made from the i-th stop to the j-th stop of test set trip k, note that in this case the model has been updated with data from the i-th stop before making the forecast
- $y_{j,k}$, the observed label of the passenger load at the j-th stop of test set trip k

Note that the stops are indexed by their respective step in the trip. The first time step $i = 0$ refers to the initial step[1], which is before the first stop of the trip. At this time step, the model has been fitted but not yet updated with real-time signals.

The first metric that we propose is the Look-ahead Absolute Error (LAE) which measures the mean absolute error based on a given forecasting horizon. The intuition is that the error of the forecasting model partially depends on how far ahead it needs to forecast. A good forecasting model will have low look-ahead errors, even for large horizons. However, it is expected that for most models the LAE increases over larger horizons. It is mathematically defined for a given horizon h as follows:

$$\text{LAE}(h) = \frac{1}{n}\sum_{k=0}^{n}\frac{1}{m-h}\sum_{i=0}^{m-h}|\hat{y}_{i,i+h,k} - y_{i+h,k}| \tag{3.7}$$

The second metric measures the improvement in error for a target stop when a single update is performed. It thus measures how incoming signals improve the model's performance. Given a target stop j, it takes the mean of the improvement in absolute error upon performing a single update, for all preceding updates. A low improvement score does not necessarily mean that a model has a low performance. If a model already has a very high goodness-of-fit, then little improvement is possible and a low improvement score will be the result. Note that for this metric a negative value indicates an improvement in the model, as the absolute error has decreased. Mathematically, it is defined as follows:

$$\text{IMP}(j) = \frac{1}{n}\sum_{k=0}^{n}\frac{1}{j}\sum_{i=0}^{j-2}|\hat{y}_{i+1,j,k} - y_{j,k}| - |\hat{y}_{i,j,k} - y_{j,k}| \tag{3.8}$$

Finally, the evaluation is not limited to these two metrics. Individual trips and forecasts are considered as well in the evaluation to understand the model's performance. Other factors such as the runtime of the model are considered as well but fall outside of the scope of the current work.

[1] In the evaluation, this will be referred to as the "START" state.

4

Data Engineering

We have previously noted that integrating the various data sources and extracting relevant features to be passed to models as input is a non-trivial task. In this chapter, we will provide a detailed description of the data engineering methods used to construct a comprehensive dataset for fitting and evaluating models that make passenger load predictions.

4.1. Overview

As mentioned previously, this research is set in the context of passenger load estimation for tram vehicles in the Rotterdam – The Hague Metropolitan area. In order to be able to fulfil the research objectives, a dataset is required that allows for fitting and evaluating models. To create this dataset, a pipeline is set up that extracts the required features and transforms the data into a tabular format. While the current pipeline is constructed for a specific context, the same techniques could be applied in other scenarios given that the same data sources are available. For the problem at hand and the creation of a dataset to address the research questions, three data sources are relevant. First, AFC data is required to create ground-truth labels of passenger load and to construct historical passenger load and flow information. Second, GTFS information is required to acquire information about the public transport network and the timetable, such as the location of certain stations. Finally, vehicle data is required to extract features as input to the models. The overall pipeline has been based on the methodology proposed by Luo et al. [8], in particular, the preprocessing steps to handle faulty AFC data.

4.1.1. AFC Data

Within the Netherlands, public transport fare collection is consolidated through the *OV-chipkaart*[1] system. Travellers store credit on a smart card which may be used to check in and check out of any public transport vehicle. For the purposes of this research, a dataset is provided that contains aggregated AFC data for all tram trips in the period 2020 and 2021 operated by the current operator. The data is collected at a stop level, where information about the trip, vehicle and passenger flow is provided for each stop in a trip. Note that this data does not contain a direct value for the passenger load in the vehicle. As a consequence, the passenger load needs to be derived based on the boarding and alighting passenger counts. An exploratory data analysis of the AFC data can be found in Appendix A. A preliminary overview of the distribution of the passenger load is provided by Figure 4.1 to provide an intuition about the spread of the data, this figure is also provided in the exploratory data analysis.

4.1.2. GTFS Data

General Transit Feed Specification is a data specification that allows public transport providers to publish transit information in a common format[2]. It can be provided statically or through a real-time stream, depending on the use case. The Dutch GTFS data is publicly available online[3] and provides information about all the scheduled trips as well as the specific names and locations of stops along routes.

[1] https://www.ov-chipkaart.nl/
[2] https://gtfs.org/
[3] http://gtfs.ovapi.nl/nl/

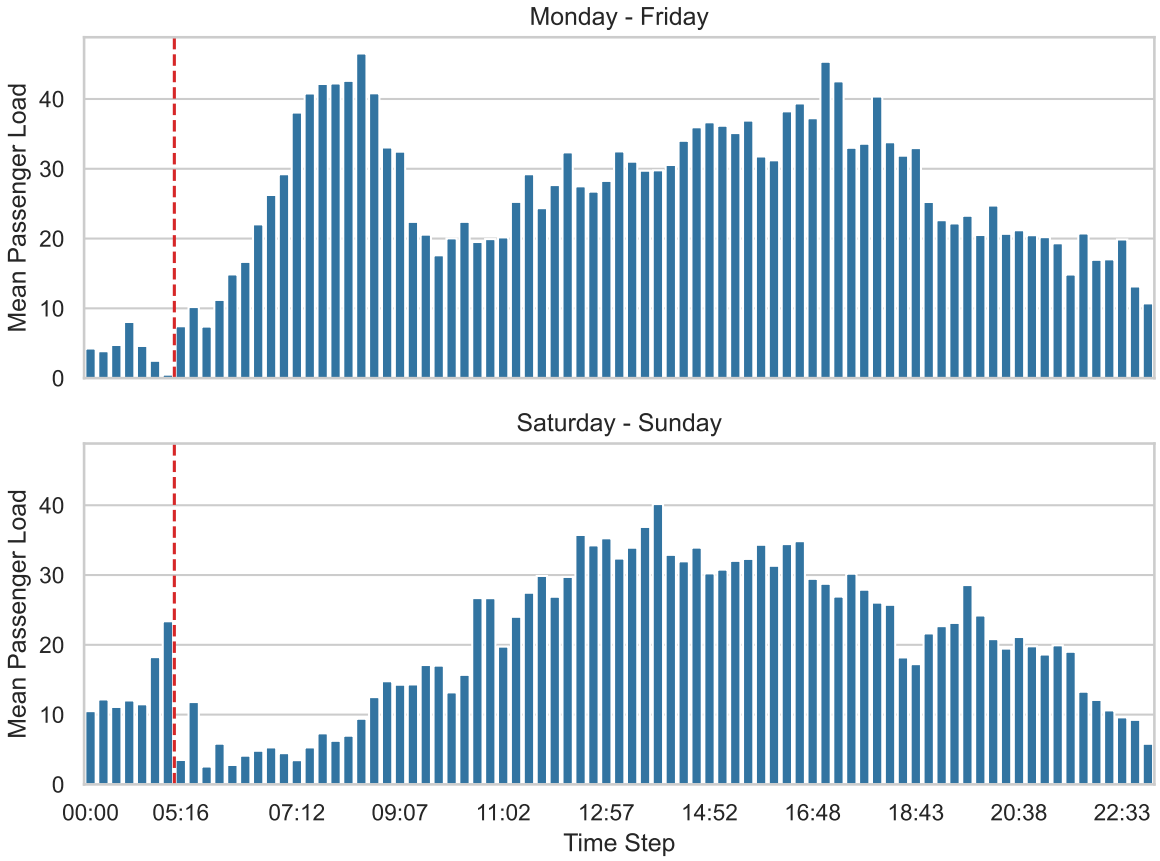

Figure 4.1: Barcharts of the mean passenger load across all trips at given time steps of the day. The upper plot shows the passenger load distribution on weekdays and the lower plot shows the passenger load distribution during the weekend days. Note that there is no public transport between roughly 1:26 and 5:16, these time steps are excluded from the figure. The gap is indicated by a vertical red line on the figure.

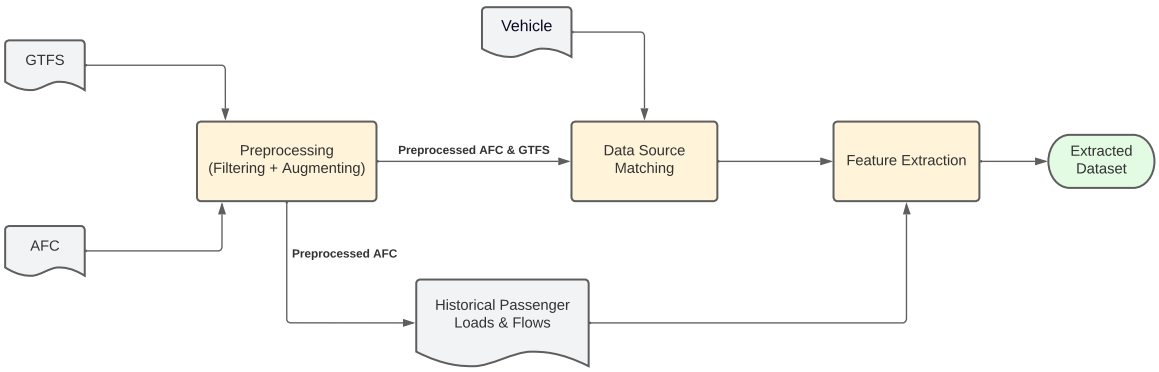

Figure 4.2: Schematic diagram of the proposed data engineering pipeline. Data sources are displayed in grey blocks, intermediary steps in the pipeline in yellow boxes and the final output in a green pill.

The available data contains the latest static GTFS information and unfortunately, little archived data is available. Therefore, when considering vehicle trips in the past there may be some discrepancies due to changes in the timetable.

4.1.3. Vehicle Data

The vehicle data is provided through an AWS S3 Bucket[4] which is constantly updated by the sensor signals of the vehicle. The signals are extracted by a computer in the tram that listens in on the multifunction vehicle bus (MVB) in the vehicle. Signals can be recorded with a frequency of up to 5 Hz. Then, the signals are uploaded to the storage bucket. The data is formatted as one row per signal value and is not grouped in any format. Therefore, significant preprocessing is required to turn it into a usable format for a model. The hardware for extracting signals has been installed on ten Avenio tram vehicles and has been in operation since 2019.

4.2. AFC & GTFS Preprocessing

Both the AFC and GTFS data require preprocessing before the data can be used in the data engineering pipeline. In both cases, information that is not relevant to the current context needs to be filtered out. That is, it should only contain information related to the lines, trips and vehicle numbers that record the data. Besides filtering out irrelevant data, the AFC data needs to be augmented by calculating the passenger load for each row in the data collection.

4.2.1. Filtering

First, both the AFC and GTFS data collections are filtered such that only relevant data is used in the pipeline. For the static GTFS data, this means that all data not related to the current tram operator can be discarded. This is straightforward, as the structure of the data allows the information to be related to a specific operator.

The GTFS and AFC collections contain information related to the trips in the current context, but also contain information regarding trips made by all other vehicles in the Netherlands. The specific tram vehicles that are under consideration are only operated on five specific tram routes: 2, 9, 11, 15 and 17. Data that is not related to those lines can be discarded. Note that in the AFC collection data related to vehicles that do not produce vehicle data is retained, as this data may be used to calculate historical passenger load and passenger flow information.

4.2.2. Passenger Load Calculation

While the passenger flow, in terms of boarding and alighting passenger counts, is provided in the AFC file, the passenger load need to be derived.

This is a non-trivial task since the data may be erroneous in some cases or the passenger load could

[4] https://aws.amazon.com/s3/

be affected by travellers forgetting to check out, hence ending a trip with a net positive passenger load. Moreover, errors in the derived passenger load value are propagated throughout the series of stops in the trip. Therefore, a method for deriving the passenger load needs to be able to handle faulty or missing data. Here, we present a method that adheres to defined constraints to facilitate the calculation of the passenger load.

Let the passenger load at a time step t in a vehicle v be defined as $l_{v,t}$. The passenger load is subject to several constraints, which may in turn facilitate its calculation. The first constraint is that the passenger load in a vehicle is never negative:

$$l_{v,t} \geq 0 \tag{4.1}$$

Second, the passenger load at the initial time step and at the final time step is also zero:

$$l_{v,0} = 0 \wedge l_{v,t_{\text{max}}} = 0 \tag{4.2}$$

Third, the passenger load can never exceed the maximum capacity c_v of vehicle v:

$$l_{v,t} \leq c_v \tag{4.3}$$

Finally, the passenger load can be defined as being equal to the cumulative sum of the board counts ($b_{v,t}$) subtracted by cumulative sum of the alight counts ($a_{v,t}$) of all previous time steps:

$$l_{v,t} = \sum_{t'=0}^{t} b_{v,t'} - \sum_{t'=0}^{t} a_{v,t'} \tag{4.4}$$

Alternatively, the calculation rule can be recursively defined as:

$$l_{v,t} = l_{v,t-1} + b_{v,t} - a_{v,t} \tag{4.5}$$

where the constraint in Equation (4.2) applies for the first time step $t = 0$.

By applying Equation (4.4), the passenger load can be calculated for each day and vehicle. However, the data is subject to errors and may contain missing rows. Therefore, the other constraints may be used to verify whether the calculation is indeed correct. Note that we did not define what the initial and final time steps referred to, whether those refer to the start and end of a trip or the start and end of a day. We will investigate this in the following section.

4.2.3. Capturing Data Errors

An intuitive assumption is that the passenger load at the beginning and end of a vehicle's trip is zero. In a theoretical sense, this is true as there are no stops outside of the trip where passengers can board or alight the vehicle. However, in practice, it may occur that a vehicle ends its trip and immediately initiates a new trip. Passengers may choose to remain seated if their itinerary continues on the subsequent trip. This assumption may thus not necessarily hold in practice and, as will be shown in this section, seems to be a false assumption in the current context. The alternative to this assumption is that the passenger load in a vehicle is zero at the start and end of the operation on a given day. In this case, the start and end time steps $t = 0$ and $t = t_{\text{max}}$ refer to the first and final stops of a day.

Nonetheless, strict calculation of the passenger load by means of Equation (4.4) may result in violations of the other constraints due to missing or erroneous data. Take, for example, the situation where a passenger alights from the vehicle but forgets to check out or simply a missing row of data that includes the information about the board and alight counts at a certain stop. To mitigate these errors, we may apply the constraints described in equations (4.1), (4.2) and (4.3).

The current approach is as follows: the passenger load is computed by applying Equation (4.5) iteratively throughout the day. If any of the constraints is violated at the beginning of the calculation for a new trip, the calculated load value is reset to zero. Moreover, if the starting passenger load of a trip exceeds a small bound of five passengers, the load value is reset as well to counterbalance the growing number of possible passengers that forgot to check out.

In order to verify that the current approach is sound, it is compared to the intuitive method based on starting every trip with zero passengers. Let the current approach be referred to as the day-based

method and the other approach as the trip-based method. Both approaches are applied to a subset of the AFC dataset, which is all the AFC data from 2020 produced by the ten tram vehicles that also provide vehicle data. This yields a set of 1,125,722 AFC data rows. The number of constraint violations is calculated for each method and the results are compared to determine the most optimal method and evaluate the general effectiveness of each method. The results are shown in Table 4.1.

The results demonstrate that the day-based approach yields roughly seven times fewer constraint violations than the trip-based approach. Therefore, it is concluded to be a suitable approach for calculating passenger loads, also taking into account the relatively low number of constraint violations. The trips with constraint violations in the calculated passenger load are discarded from the dataset after preprocessing.

4.3. Historical Passenger Load and Flow

An important feature in the literature for predicting passenger load or flow values is some form of historical data correlated with the output. Here, we refer to both historical passenger load and passenger flow data since both could be useful in predicting the passenger load.

In this work, the historical data is defined as the average of all load and flow values at the same stop at the same time step, for all trips on the same route. In other words, the historical passenger load, board count and alight count are defined as the average of all previous values at the same time of day for similar trips. A concrete definition of the time step is provided in Section 4.5.5.

In addition to these historical features, which provide a daily average. Historical features are also extracted at a weekly and monthly level. That means that the same history is gathered but it is restricted to be on the same day of the week or day of the month. In this manner, passenger load and flow patterns at three different temporal levels can be provided to the predicting model.

The data used to extract this historical information is all the AFC data that is available on the relevant routes, not only the AFC data by the vehicles that also produce the vehicle data. Moreover, as only data for a specific period is available (2020 and 2021), history is gathered over the whole range of data, even if that means that the *history* is in the future of a given instance. This is to ensure that enough data is available to create these historical features for the earlier instances in the dataset. This is not expected to have an influence on the generalisability of the results. In a real-time setting, the historical data would self-evidently be collected from actual historical data.

Overall, the historical data provides nine features to the model: passenger loads, board counts and alight counts, each on a daily, weekly and monthly level.

4.4. Matching Data Sources

As described previously in Section 4.1, three data sources are used to construct the desired dataset. Combining these three is a non-trivial task. In this section, we describe the method employed to match data sources. Due to the complexity of the procedure and it being the basis for the construction of the dataset for evaluation, it is described in detail.

First, it is important to define the terminology in the current context. We are considering tram vehicles operated by the current tram operator. Each vehicle traverses the network over one of several *routes* or *lines*[5], these two terms may be used interchangeably and refer to the same concept. The instantiation of a route in a certain direction at a certain time moment is referred to as a *trip*. The operator assigns an id to each trip in the schedule. Note that this *trip id* is not unique, the same id may be assigned to trips on different days. In fact, most of the trip ids are repeated every day for the same trip at a certain time of day until the timetable is changed. Hence, a certain trip is uniquely identified by its id and the

[5]As mentioned in Section 4.2.1, the routes under consideration are: 2, 9, 11, 15 and 17.

	Constraint Violations	Proportion w.r.t. Total Rows
Day-based	12,645	0.011
Trip-based	86,242	0.077

Table 4.1: Constraint violations in the 2020 AFC dataset when calculating the passenger load after applying both the day-based and trip-based passenger load calculation methods.

date of execution. To avoid duplicate trip id's between operators, the GTFS defines separate unique id's for all trips. During a trip, the vehicle stops and passes several *stations* according to the defined *route*. The act of passing a station is called a *stop* and the traversal of the route between two stations is referred to as a *link*.

The preprocessed AFC data contains trip id's, scheduled departure times and vehicle id's. Hence, this dataset forms the best basis for matching the other datasets. Besides a scheduled departure time, the AFC data also provides an "actual" departure time. This is the moment where the check-in/-out machines configure to the next stop. It is slightly inaccurate but is close to the true departure time. Based on a post hoc analysis, in 95% of the cases this timestamp is later than the true departure time.

The matching occurs at the stop level, at which the data sources can be aligned. The AFC data is matched to the GTFS by the name of the station and the trip id or line number and consequently, the vehicle data is matched by the GPS location of the station retrieved from the GTFS and the departure time provided by the AFC data.

First, we consider the matching of the AFC data to the GTFS data. If the trip id exists in the GTFS data, then the station information is retrieved using the trip id and station name combination. However, it may occur that the trip id no longer exists in the GTFS. In those cases, the line number and station name are used to retrieve information. If that combination does not work either, the station name on its own is used to retrieve the station information. Note that the latter two methods are slightly more inaccurate and may also yield several matches. This occurs because information for a station may be multiply defined for different trips. Consequently, in those cases, there could be a discrepancy between the given scheduled departure time and the GTFS scheduled departure time or the returned GPS location. Therefore, the first method of retrieval by the trip id and the station name is preferred and the other two methods are used as fallback methods. If multiple matches are found, then the match with a matching scheduled departure time is used. Otherwise, a random match is selected.

The GTFS data provides the GPS location of each specific station, even differentiating between platforms on different sides of the road. This can be used to match the vehicle data to the stop. If the vehicle, whose id is given in the AFC data, is within a certain range of the stop within a certain time frame and it has stopped, then the exact arrival and departure time for that stop can be calculated and all three data sources have been synchronised for that stop. The value of the range should be large enough so that even if the GPS location in the GTFS is inaccurate, the vehicle is still within range but small enough such that it does not overlap with other stops either. We determine this value empirically by calculating the distances between all consecutive stops as defined by the relevant routes of the 2020 and 2021 AFC collection. A histogram of the computed distance values is shown in Figure 4.3. Note that some stations are narrowly separated. This usually occurs at the beginning and end of a trip where a starting station or ending station is processed twice in the AFC data to denote the start or end of a trip. Ignoring these occurrences, the minimum station distance is 138.29 meters. Accounting for inaccuracies in the GPS data, the maximum range is set at 100 meters.

In some situations, no GPS matching can be successfully performed. Most commonly, two exceptions occur in the data: there is no valid GPS data available or there is valid GPS data available but the vehicle did not stop when it was in range.

The first situation occurs either when the GPS equipment in the vehicle malfunctions or when the tram is below ground. In The Hague, several stops in the city centre are below ground and hence this situation occurs frequently. In such cases, the only way to match the vehicle data to the AFC data is to use the "actual" departure time value and find the closest time frame where the vehicle was stopped to that value.

The second situation occurs when the vehicle passes a station without stopping. This could happen when the driver notices that nobody is going to alight at the upcoming stop and also sees that nobody is waiting to board at the station. In that case, the driver may decide to not stop at the station and proceed to the next station. To ensure that this is actually the case, the AFC row should have zero boarding passengers and zero alighting passengers. However, in some rare cases, the vehicle did not stop at the station and the board count is zero but the alight count is not zero. Then it may still be the case that the train did not stop but that the alighting passengers checked out too early with their smartcards for the succeeding stop. In fact, the card reader takes several seconds to update for the next stop after passing a stop so the alighting passengers may intend to check out for the next stop but accidentally check out too early. This, again, relates to the fact that the provided "actual" departure time in the AFC data is often later than the true departure time.

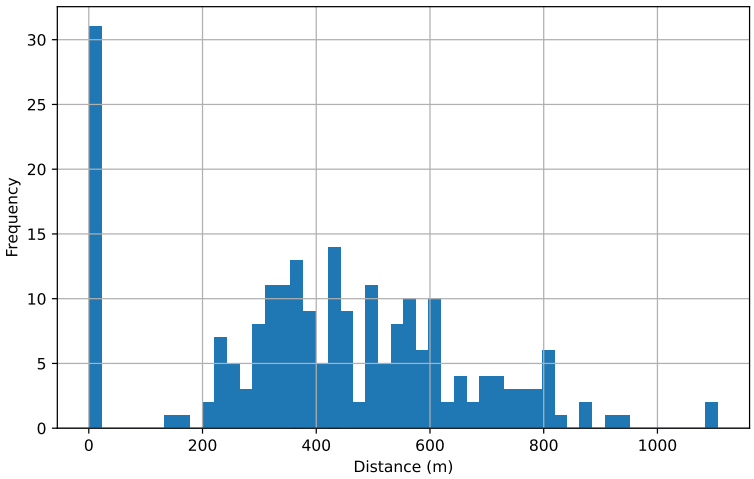

Figure 4.3: Distribution of station distances for all unique pairs of consecutive stops in the 2020 and 2021 AFC collection, excluding 220 values with a distance larger than 1500 meters.

Despite taking these exceptions into account, several rare cases may still occur where no data matching can be successfully done. Those stops are ignored in the data engineering pipeline.

4.5. Feature Extraction

In this section, we will describe the specific methods of calculating or aggregating the source data into feature values. Most of the features originate from the vehicle data but several are derived using the AFC data or the GTFS data. Features that required non-trivial aggregation are explained in detail and other less-complex features are enumerated at the end of this section.

4.5.1. Weight Estimate During Acceleration

The weight estimate is one of the most complex features in the data engineering pipeline. Coincidentally, it is also one of the most crucial features as will be demonstrated in Chapter 5. By definition, the weight of the vehicle has a direct and linear relationship to the number of passengers in the vehicle. Therefore, having a measurement of the weight may be crucial for a model that estimates the number of passengers. Unfortunately, the vehicles in the current context do not measure this feature. However, an approximation of the weight can be made by means of other sensor signals, namely the produced force and speed measurements in the period that the vehicle departs from a station.

Newton's second law of motion defines Force (F) as the product between mass (m) and acceleration (a):

$$F = ma \tag{4.6}$$

Using this equation, the mass can be calculated as such:

$$m = \frac{F}{a} \tag{4.7}$$

Hence, if the produced force is known as well as the acceleration, then the mass (or weight)[6] of the vehicle can be calculated.

Fortunately, the produced force is measured in the vehicle. The Avenio tram vehicle contains three so-called traction control units. The traction system converts the electrical energy collected from the overhead catenary via the pantograph into mechanical energy, which allows the wheels to turn and move the vehicle. At each unit, the produced force is measured in Kilonewtons (kN). The acceleration can be calculated using speed signals.

[6]Equation (4.7) describes the calculation of the *mass* of the vehicle. However, in this work we elect to refer to this property as the *weight* of the vehicle for better reading purposes.

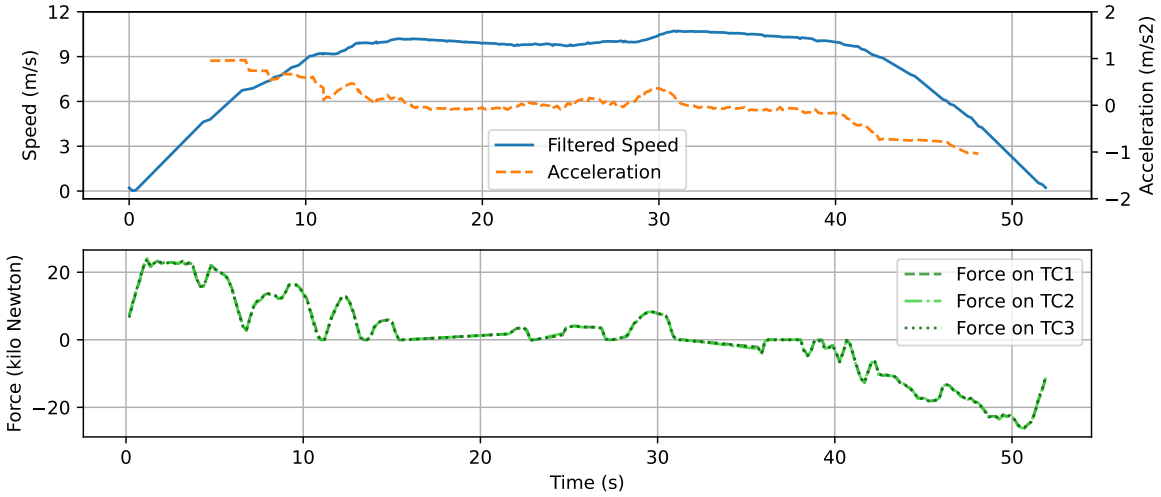

Figure 4.4: Speed, acceleration and exerted force signal values of an Avenio tram vehicle on 15 September 2020 at 15:22. Speed signals have been filtered due to signal noise in the data, and the noisy data points have been removed.

In order to get a good estimate of the weight, the calculation is made when the vehicle accelerates from a stopped position to a regular speed, in essence as soon as the acceleration is no longer larger than zero. This occurs after departure from the station after a stop. For the final estimate, one needs to aggregate multiple signal values into the F and a values in Equation 4.7. For the force parameter F this will be the sum of the average exerted force over the acceleration period for all three traction control units in the vehicle. For the acceleration a, this will be the difference in speed divided by the length of time. Hence, given a window of data starting at time step $t = 0$ and ending at $t = n$, the weight estimate w' is calculated as follows:

$$w' = \frac{\overline{F}_{\text{TC1}} + \overline{F}_{\text{TC2}} + \overline{F}_{\text{TC3}}}{\frac{s_n - s_0}{t_n}} \tag{4.8}$$

where $\overline{F}_{\text{TC}x}$ is the mean force exerted by traction control unit x in Newtons (N) over the entire period, s_t is the speed at time step t in meters per second (m/s) and t_n is amount of elapsed time at $t = n$ in seconds (s). An example of what the signal values look like on the link level, between two stations, is given by Figure 4.4.

While the estimate can be calculated by taking the average exerted force as well as the acceleration over the complete time window, one could also make various estimates using sliding windows over the complete period of acceleration, or on a predefined sub-window of the entire period. Note that due to a misconfiguration, the speed signals contain a noisy signal every second. These are removed before processing and calculation of the weight estimate. An experiment is set up to evaluate which method of weight estimation is optimal. In this case, we identify three main methods of finding the calculation window:

1. Taking the entire period of acceleration as the window
2. Finding a period of stable acceleration and taking that period as the window, in this case, a ”stable” is defined as the period where the acceleration does not change by more than 0.05 m/s^2
3. Taking the period where the speed is within a defined range, for example, 5 km/h to 10 km/h

Moreover, each period can again be divided up into intervals of a certain time length. An estimate can be made for each interval and the mean of the estimates can then become the final estimates. A dataset is collected of 10,536 individual stops over 353 vehicle trips including data for each of the ten vehicles. For each stop, different weight estimation methods are applied. As there is a linear relationship between the vehicle weight and the passenger load, the Pearson correlation coefficient of

Weight Estimation Method	Correlation to Passenger Load
Full window, 500 ms interval	0.133
Full window, 1000 ms interval	0.204
Full window, 1500 ms interval	0.237
Full window, 2000 ms interval	0.175
Full window, 3000 ms interval	0.319
Full window, 4000 ms interval	0.368
Full window, no interval	0.407
Stable window, 500 ms interval	0.112
Stable window, 1000 ms interval	0.185
Stable window, 1500 ms interval	0.194
Stable window, no interval	0.203
Window 0 km/h to 5 km/h, 500 ms interval	0.108
Window 0 km/h to 5 km/h, no interval	0.175
Window 5 km/h to 10 km/h, 500 ms interval	0.167
Window 5 km/h to 10 km/h, no interval	0.114
Window 10 km/h to 15 km/h, 500 ms interval	0.103
Window 10 km/h to 15 km/h, no interval	0.141
Window 15 km/h to 20 km/h, 500 ms interval	0.087
Window 15 km/h to 20 km/h, no interval	0.117

Table 4.2: The Pearson Correlation of each respective weight estimation method with respect to the passenger load in the vehicle. The results have been calculated on a set of 10,536 individual vehicle stops where a window of sensor signals is collected at the time of departure until the vehicle stops accelerating.

each method to the passenger load can be used to determine the most optimal method, see Equation (3.1) for its definition. Table 4.2 shows the results of this analysis.

The results indicate that simply using the entire acceleration period as a window yields the highest correlation. Hence, this is the method that is used to extract the estimated weight as a feature in the data engineering pipeline.

4.5.2. Door Open Times

The Avenio tram has ten doors, five on each side of the vehicle. The length of time that the doors are opened can be used as a feature for the estimation models. The signal that indicates whether an individual door is open is a binary signal that is flipped when the door is opened or closed. An example of these signals is shown in Figure 4.5. The time period between the signal that the door is opened and the signal that the door is closed again can be used to determine the door open time. This will yield one feature for each of the ten doors. It could occur that doors are opened more than once during a stop. In that case, the sum of the periods is taken as the door open time.

In addition to these door open time features, the total sum of all door open times at a stop is extracted as a feature as well. It is essentially a summary feature of the individual door features.

Finally, the amount of door open/close cycles at a stop is also captured in a single feature, where the value is the total amount of open/close cycles of all the doors at a stop. Hence, if all five doors on one side have been opened once at a station, the value of this feature would be 5.

4.5.3. HVAC and Temperature Features

Climate control is an important component of many modern public transport vehicles and its configured state is undoubtedly affected by the number of passengers within the vehicle. To facilitate climate control within the Avenio tram vehicles, there are four Heating, Ventilating and Air Conditioning (HVAC) units. The sensor signals produced by these units provide the opportunity to extract several features. The intuition is that, given that humans produce heat and humidity, the state or mode of the HVAC units may give insight into the passenger load. Consider a scenario where the vehicle is completely full, then the HVAC units need to cool and dehumidify significantly to maintain a comfortable climate within the vehicle. On the other hand, when the vehicle is (almost) empty, then the HVAC units need to work less hard to maintain the climate.

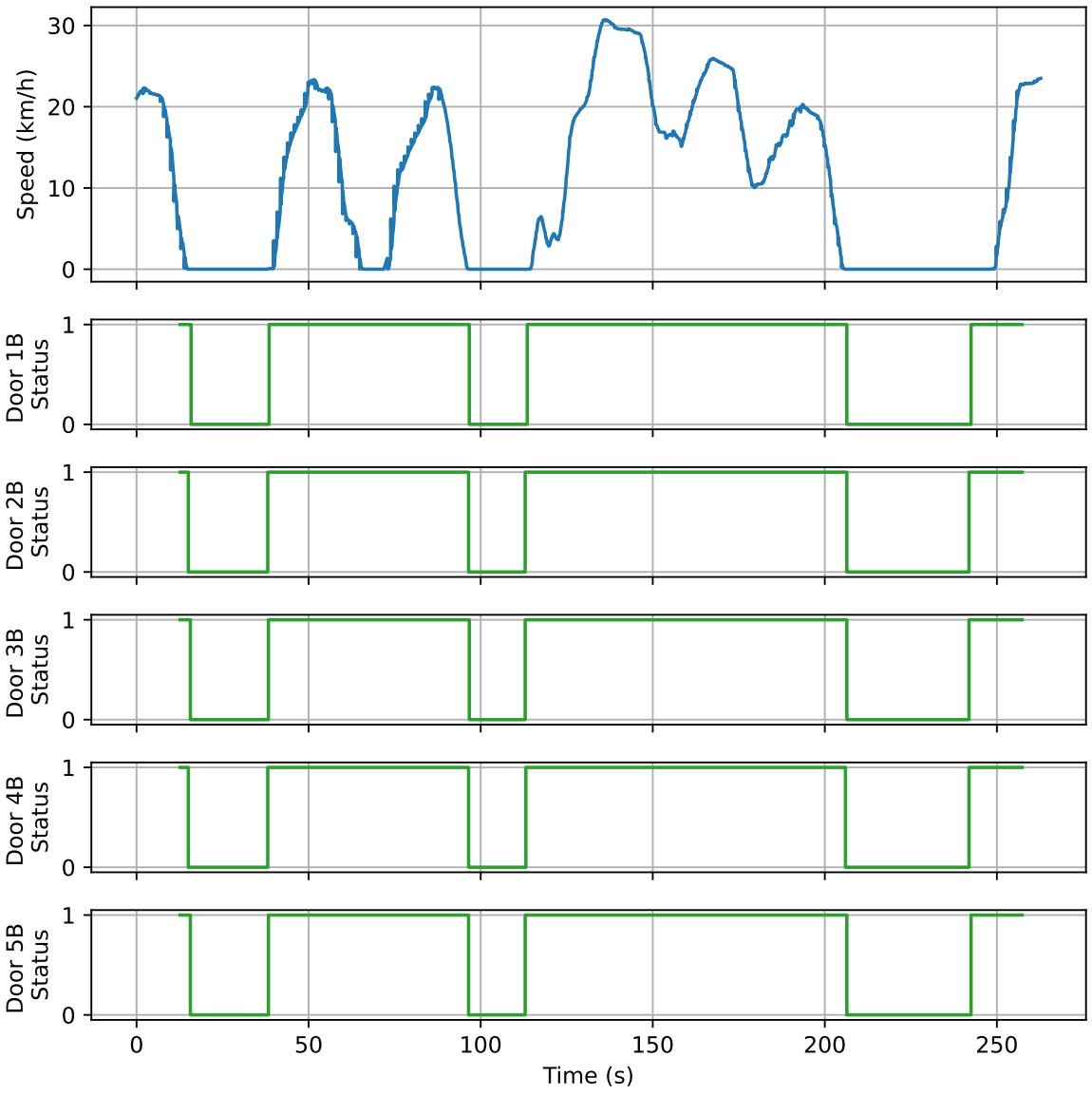

Figure 4.5: Speed and door signals (B-side of vehicle: right side) of one of the Avenio vehicles on 10 June 2020 between 21:19 and 21:23. Note that the noisy speed signals have not been filtered out in the current figure.

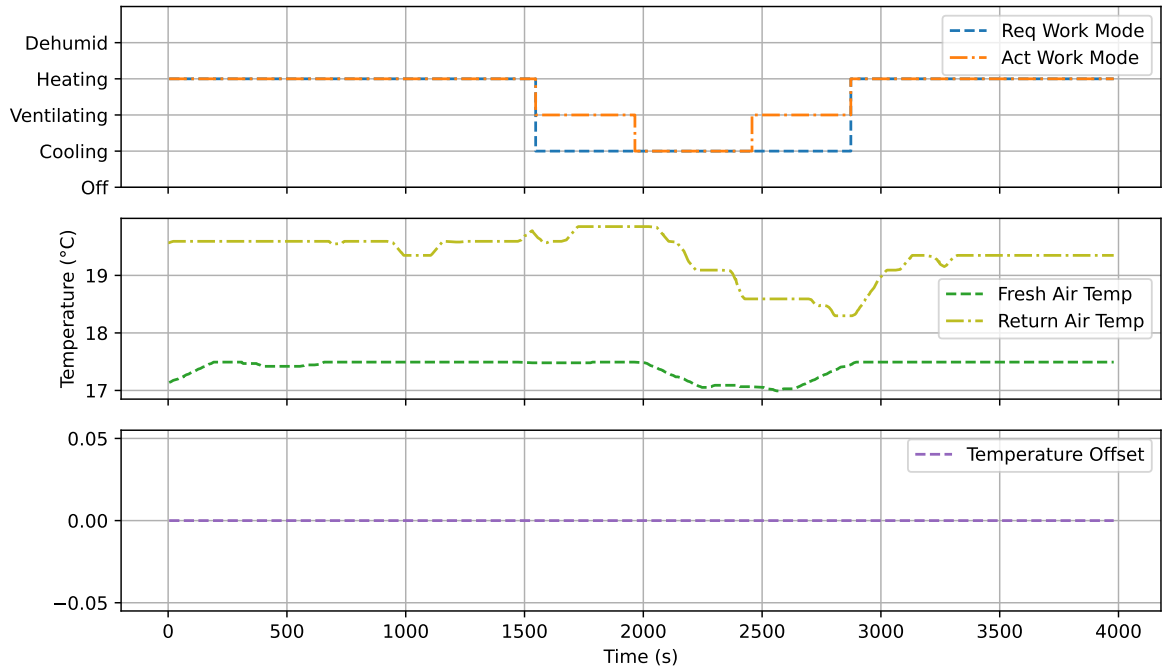

Figure 4.6: HVAC signal values of HVAC unit 2 of an Avenio vehicle on 30 June 2020 between 16:47 and 17:53. The required and actual working modes have been abbreviated to “Act” and “Req” and the temperature has been abbreviated to “Temp” in the figure.

The four HVAC units each have a number: 1, 2, 4 and 5. There is no HVAC unit 3 in this specific model of the Avenio tram, therefore it is not present in the data either. HVAC units 1 and 5 are located in the two driver cabins at the outer sides of the vehicle and HVAC units 2 and 4 are located in the passenger cabin.

Each HVAC unit provides its own signals. The feature extraction will focus on five signals: the HVAC unit’s actual working mode, its required working mode, the fresh and the return air temperatures and the VCU temperature offset. Both working mode signals are nominal values, indicating one of the five possible working modes: dehumidifying, heating, ventilating, cooling and off. The difference between the required and actual working mode is that the former indicates the configured working mode and the actual indicates the current state of the HVAC unit. As the actual working mode is more representative of the state of the vehicle, this signal will be used as a feature rather than the required working mode.

The fresh air temperature signal measures the temperature of the incoming air and the return air temperature signal measures the temperature of the outgoing air of the HVAC unit, both are in degrees Celsius (°C).

Another signal that is of note is the temperature offset which can be tweaked by the driver. This is a parameter that affects the offset of the required cabin temperature from the normal temperature. If that value is negative, the passenger cabin will be slightly colder than the usual temperature. This feature is also included in the model. It may vary between -3 and 3 but is most often set at 0.

An example of the signal values is shown in Figure 4.6. Note the effect of the working mode on the return air temperature in particular.

The features are collected on the link level and for each individual HVAC unit. For the working mode, the most frequent signal value in the relevant time period is used as a feature. The feature extracted from the air temperatures is the difference between the mean return air temperature on the link and the mean fresh air temperature on the link. This indicates the degree of cooling/heating by the HVAC unit. A positive value of the temperature delta indicates that the return air is warmer than the incoming air and a negative value indicates that the return air is colder than the incoming air. This is referred to here as the temperature delta of the HVAC unit. Finally, for the temperature offset value, the most frequent value is used as well.

4.5.4. Headways

Another feature of note, also considered in the literature [4], is the headway between vehicles at a stop. The headway is the delay between the arrival of succeeding vehicles at a stop. Here, we differentiate between the headway between a vehicle at a stop and any other vehicle at the same stop and the headway between a vehicle at a stop and a vehicle of the same line and direction at a stop. This information can be retrieved using the static GTFS data. For a given stop, the associated GTFS information is retrieved and all preceding stops by other vehicles. Then both headway features can be directly calculated by taking the difference in arrival time of both stops in seconds. In the case that the given stop is the first stop of the day, no preceding stops will be found. In that case, no value for these features is returned. Note that only the *scheduled* headway is taken into account rather than the *actual* headway, which also takes delays into account. Retrieving the actual headway is not possible as the actual arrival and departure times of other vehicles cannot be determined to the same degree of accuracy and will thus yield an inaccurate feature value.

4.5.5. Miscellaneous Features

This section will describe the less-complex features in the data engineering pipeline which can be extracted directly from the data. These are all computed on the stop level.

- **Trip Progress**: the progress of the vehicle along its trip, where 0 means the trip has yet to begin and 1 means the trip has been completed
- **Lateness**: the delay of the vehicle compared to the schedule, the difference between the actual arrival time and the scheduled arrival time in seconds
- **Dwell Time**: the time spent stopping at the station, the difference between the departure time and the arrival time in seconds
- **Time Step**: a value ranging between 0 and 100 indicating the rough time of day of the departure time of the vehicle (as proposed by Ding et al. [38]), starting at 00:00 am and ending at 23:59 pm. Every time step constitutes roughly 15 minutes.
- **Day of Week**: a number indicating the day of the week of the current departure time of the vehicle from the stop, where 0 indicates Monday and 6 indicates Sunday (as proposed by Ding et al. [38])
- **Day of Month**: similar to the day of the week, indicates the day of the month (as proposed by Ding et al. [38])
- **Is Outer Station**: a boolean value that indicates if the current stop is either a start or an end station of the trip
- **Line Number**: the line number as provided by the AFC data through the trip id

4.6. Discussion

We have presented the methods used for extracting relevant features from the given heterogeneous collection of data. Three main data sources are used: AFC, GTFS and vehicle data. A structured pipeline has been constructed that consolidates the data of each of these sources. Using that pipeline, a wide variety of features has been defined that can be incorporated into a dataset for a predictive model to use for real-time passenger load estimation, which is one of the core objectives of this work.

In order to ensure the correctness of the passenger load data, we have defined a set of constraints that ensure that the calculated passenger load values are optimally accurate. Historical passenger load and flow patterns are extracted as features by aggregating the average load/flow values on a particular temporal scale: daily, weekly and monthly. Vehicle features are collected by aggregating sensor signals over selected time periods. In some cases, the signals of multiple sensors are summarised into a single feature. This has been done for the door open times. Besides the features of the individual door open times, a total door open time feature is defined as well. In cases where the optimal method for extracting the feature was not obvious, the method was selected through empirical evaluation. This has, for instance, been done for the weight estimate feature. A complete overview of the features can be found in Appendix B.

The methods proposed in this chapter can be used to extract features from sensor signals from more vehicles than the Avenio tram vehicles under consideration. If any provided tram vehicle captures similar sensor signals, the currently proposed methods can be reproduced. It is, however, important to take the accuracy and granularity of this sensor data into account. For certain features, such as the weight estimate, a relatively high-frequent and accurate set of sensor signals is required in order to derive a reasonable estimate. The current maximum frequency of the sensors is 5 Hz. For a weight estimate, such a frequency is desirable due to its relatively short calculation window. However, for other features such as the HVAC temperature deltas, a lower frequency may still be satisfactory for a proper feature value. When performing such reproduction, these factors are important to take into account.

5

Feature Analysis

One of the research objectives as stated in Section 1.2 is to find out which features available from the in-vehicle data have a strong predictive power for the passenger load in that vehicle.

Each feature may be related to two factors: the passenger load and the passenger flow. To reiterate, the passenger load is the absolute number of passengers in the vehicle at a given moment and the passenger flow is the number of boarding and alighting passengers at a given stop. Features that have a relationship to the passenger flow tend to be measured at the stop level, which is where the flow of passengers occurs. On the other hand, features that relate to the passenger load tend to be measured at the link level (i.e., between two stops), where the passenger load remains constant.

Note that we have not considered the overall characteristics of the AFC data in this feature analysis. However, understanding overall trends in the AFC data can be useful for the model design as well. To this end, an exploratory AFC data analysis has been performed and the results of which can be found in Appendix A.

This chapter will explore the relationships that the features may have with the passenger load and flow. Besides a quantitative assessment of the features, a qualitative analysis is conducted as well which considers both linear and non-linear relationships. Finally, the results of these analyses are summarised and discussed with respect to the related research question RQ1. A complete overview of the features under consideration can be found in the Appendix B.

5.1. Quantitative Analysis

A quantitative analysis may show specific relationship types between the features and the target variables. Section 3.3 described three measures used for the quantitative analysis: the Pearson's correlation coefficient, the F-statistic (univariate ANOVA) and Mutual Information. In addition to the in-vehicle features, some additional features are considered as well. These are temporal and GTFS-related features. The features not included in the current quantitative analysis are historical AFC features, as these are expected to be very highly related to the targets. These features are evaluated separately in Section 5.3. A complete overview of all the results of the quantitative analysis can be found in Appendix C.

5.1.1. F-test and Correlation

Tables 5.1 and 5.2 show the top-15 features in terms of their relation to the passenger load and flow. Note that the top-ten features are selected based on their F value, which is computed by means of univariate linear regression. Therefore, it expresses the degree to which the features and the passenger load/flow are *linearly* related.

It is immediately obvious that the features in the tables show a *significant* linear relationship. This holds for virtually all features, as can be seen in Appendix C. It is also clear that no vehicle-related features show a strong correlation ($|r| > 0.5$) to either the passenger load or passenger flow. It can already be concluded that of these features, no single feature is strongly related in linear terms to either the passenger load or flow.

Feature Name	r value	F value	p value
weight_estimate_acc	0.40	57,114	0.0
prev_stop_is_outer_station	-0.22	15,244	0.0
prev_stop_overall_headway	-0.18	9,920	0.0
line_numer_9	0.17	8,861	0.0
hvac1_mode_2	-0.15	6,379	0.0
hvac2_mode_2	-0.13	5,278	0.0
prev_stop_line_headway	-0.13	4,908	0.0
line_numer_11	-0.13	4,851	0.0
door_cycle_count	0.11	3,626	0.0
dwell_time	-0.11	3,354	0.0
hvac1_mode_3	0.10	2,996	0.0
line_numer_17	-0.10	2,942	0.0
hvac2_mode_3	0.09	2,513	0.0
day_of_month	-0.09	2,273	0.0
hvac2_temp_delta	0.09	2,183	0.0

Table 5.1: Top-15 most strongly linearly related features to the passenger load over the whole dataset. The r value indicates Pearson's correlation coefficient, the F value denotes the F statistic and the p value denotes the F-test p value. Results for (one-hot encoded) nominal features have been denoted with the specific value for which the statistics are computed in brackets.

The strongest relationship is between the weight estimate and the passenger load with a correlation of approximately 0.4. Interestingly, the feature also has a relatively high correlation to the passenger flow albeit significantly lower than with respect to the passenger load. The overall headway seems to be a strong feature in both respects as well. The HVAC units 1 and 2 have a linear relationship for both modes 2 and 3 to the passenger load. Moreover, the line number also appears to have an effect. Notice how line number 9 is positively correlated while lines 11 and 17 are negatively correlated. For the passenger flow, it is clear that the door-related features, as expected, have the most significant relationship to the passenger flow. However, also notice the relationships of line number 9, the "is outer station" feature and the line headway.

5.1.2. Mutual Information

In a similar manner as the previous section, tables 5.3 and 5.4 display the top-15 features in terms of the Mutual Information between them and the passenger load and flow. As described in Section 3.3, Mutual Information measures a more general statistical relationship based on entropy which is not limited to linear relationships. Hence, it may reveal different types of relationships than the linear methods analysed in the previous section.

Note that the weight estimate has the strongest score, both with respect to the passenger load and flow. Observe that the top-ten features for both the passenger load and flow are roughly the same and in a similar ordering. Also observe the strength of the relationships of the HVAC temperature delta's as well as the outside temperature, which was not obvious from the linear analysis in the previous section. Again, it is also clear that door-related features have a strong relationship to the passenger flow. Surprisingly, these relationships are not as strong as the weight estimate and HVAC temperature deltas.

5.2. Qualitative Analysis

The qualitative analysis will provide an assessment of the relationship between features and both the passenger load and passenger flow. It mainly consists of a visual inspection of figures that relate each feature to either of the target variables. The analysis will be limited to the top features of the quantitative analysis in the previous section. The existence or non-existence of relationships will be discussed per feature with a reflection on their explanatory power with respect to the target. In cases where the distribution of the data is not clear from the figure, histograms or kernel density plots will be additionally provided. For continuous features primarily scatter plots will be used. For discrete or categorical features mostly violin plots will be used.

Feature Name	r value	F value	p value
door_cycle_count	0.24	18,288	0.0
weight_estimate_acc	0.21	13,743	0.0
prev_stop_overall_headway	-0.20	12,011	0.0
total_door_open_time	0.19	10,494	0.0
line_numer_9	0.19	10,390	0.0
dr_5b_open_time	0.16	7,353	0.0
dr_3b_open_time	0.15	6,659	0.0
dr_4b_open_time	0.15	6,571	0.0
dr_2b_open_time	0.15	6,285	0.0
dr_1b_open_time	0.11	3,454	0.0
dr_5a_open_time	0.11	3,412	0.0
dr_2a_open_time	0.10	3,064	0.0
dr_3a_open_time	0.10	3,059	0.0
prev_stop_is_outer_station	-0.10	2,925	0.0
prev_stop_line_headway	-0.10	2,887	0.0

Table 5.2: Top-15 most strongly linearly related features to the passenger flow over the whole dataset. The r value indicates Pearson's correlation coefficient, the F value denotes the F statistic and the p value denotes the F-test p value. Results for (one-hot encoded) nominal features have been denoted with the specific value for which the statistics are computed in brackets.

Feature Name	MI value
weight_estimate_acc	1.356
hvac5_temp_delta	1.241
hvac4_temp_delta	1.236
hvac2_temp_delta	1.203
hvac1_temp_delta	1.200
average_out_temp	1.048
trip_progress	0.372
vehicle_lateness	0.232
total_door_open_time	0.217
dwell_time	0.209
prev_stop_overall_headway	0.185
stop_time_of_day	0.131
prev_stop_line_headway	0.113
prev_stop_is_outer_station	0.077
dr_3b_open_time	0.071

Table 5.3: Top-15 most strongly related features to the passenger load based on their Mutual Information score.

Feature Name	MI value
weight_estimate_acc	0.891
hvac5_temp_delta	0.870
hvac4_temp_delta	0.867
hvac1_temp_delta	0.836
hvac2_temp_delta	0.836
average_out_temp	0.700
total_door_open_time	0.499
dwell_time	0.458
trip_progress	0.281
door_cycle_count	0.252
dr_3b_open_time	0.164
dr_5b_open_time	0.154
dr_2b_open_time	0.150
dr_4b_open_time	0.146
dr_3a_open_time	0.145

Table 5.4: Top-15 most strongly related features to the passenger flow based on their Mutual Information score.

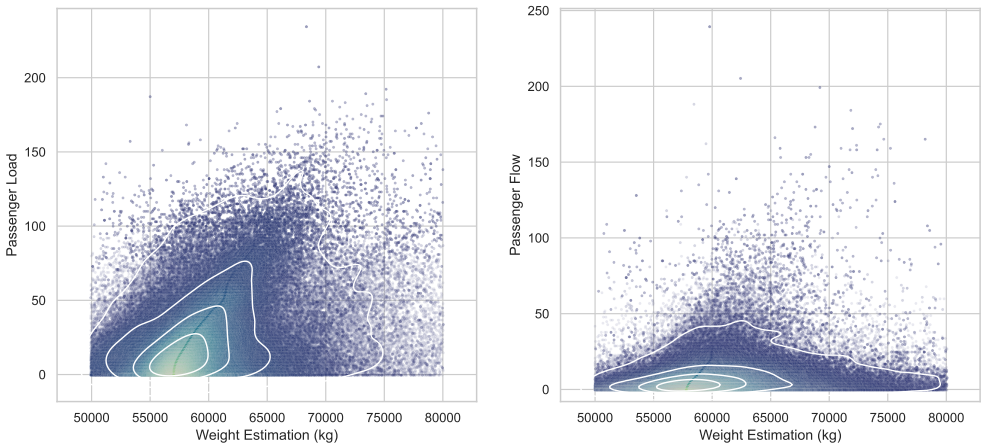

Figure 5.1: Scatter plot of weight estimate versus passenger load (left) and the passenger flow (right) including colouring and regions based on estimated Gaussian kernel density.

5.2.1. Weight Estimation

Consider Figure 5.1 which shows the relationship between the weight estimates and the passenger load and the passenger flow. Although the relationship does not seem very strong, there seems to be a relationship between the weight estimate and both targets. For the passenger load, this makes sense intuitively, as the vehicle's weight is linearly related to the number of passengers. The relationship with respect to the passenger flow is weaker, but there seems to be a slight relationship. The noise is most likely introduced by the inaccuracies in the measurement of the sensor data or the method of calculation.

5.2.2. Door Features and Dwell Times

With regards to the door open features, we will not consider the individual door signals for the qualitative analysis as these are summarised by the total door open times feature. Figure 5.2 shows the relationship between the total door open times and the passenger flow. The relationship is shown for all stops as well as only non-outer station stops. This is because, at outer stations, vehicles tend to stay idle with the doors open for a long period of time before starting a trip. The graph showing data for non-outer stations, in particular, shows a slight trend where the passenger flow is higher for longer door

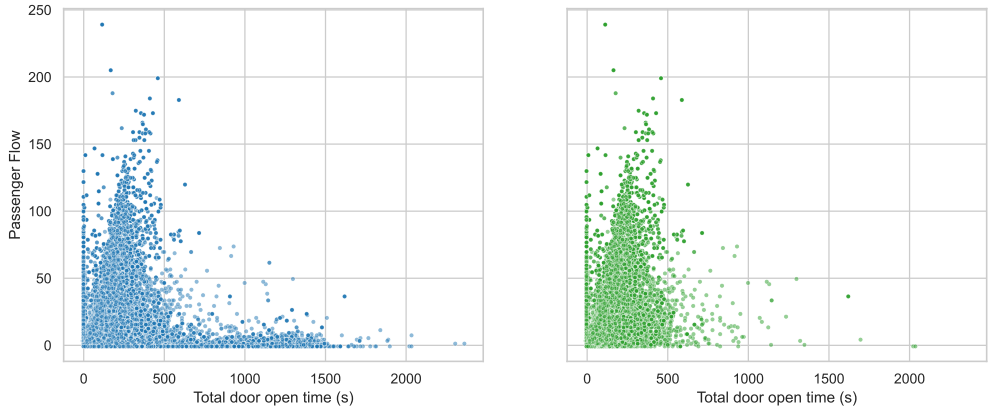

Figure 5.2: Scatter plot of total door open times versus passenger flow for all stations (left) and non-outer stations (right).

open times. Observe that there is also a vertical trend when the door times are approximately zero. It seems to indicate an error in the data or in the feature extraction process because it is not practically possible to have unopened doors but still have a non-zero passenger flow.

The door cycle counts were found to have a significant relationship to the passenger flow as can be seen in Table 5.2. Figure 5.3 shows a scatter plot of the relationship between the door cycle count feature and the passenger flow including marginal distributions. While a direct relationship is not clear, it seems that all door cycle count values which are not five do not yield a high passenger flow. However, most of the data is at a door cycle count of five. As the quantitative analysis showed a slight positive correlation, this indicates that a higher door cycle count value would lead to a higher passenger flow value. This is not obvious from the figure.

Another related feature to the door features is the dwell time feature. The dwell time is self-evidently closely related to the door open times. Therefore, we will examine the relationship with the passenger flow similarly to the door open times. Figure 5.4 shows this relation. Note that one may observe a similar trend as with the door open times.

5.2.3. Lateness

Figure 5.5 shows two scatter plots of the relationship between the vehicle's lateness and the passenger load. The majority of the data is in a region around a lateness value of zero so another plot for only that region is added as well. Observing closely, it becomes apparent that for a positive lateness value, the passenger load is positively affected. The effect is only slight but may indicate a positive relationship. There are also more outliers to the right of the region than on the left side. While the quantitative analysis did not show a positive linear relation to the passenger load, the figure does seem to indicate a slight relationship.

5.2.4. HVAC Mode and Temperature Delta

There are several HVAC features that are interesting for the qualitative analysis. Namely, these are the HVAC modes and temperature deltas. As these are measured on the link level, these are expected to relate to the passenger load. Intuitively this makes sense, as more passengers have an increased effect on the climate in a vehicle.

The first feature we consider is the HVAC mode feature. Figure 5.6 shows violin plots displaying the relation of the HVAC modes to the passenger load. The distributions are thin for high-load scenarios. Also note that distribution between each mode, the heating mode is the most common mode for all HVAC units. It is therefore difficult to say whether the fact that the heating mode occurs more frequently in higher-load scenarios is due to the value of the mode itself, or just that it is more common in general. Note that HVAC units 1 and 2 do not seem to ever have the dehumidifying mode as their value and rarely do the HVAC units 4 and 5.

Secondly, we consider the individual HVAC unit temperature deltas. To reiterate, the temperature delta is the difference between the return air temperature and the fresh air temperature. Thus, a positive

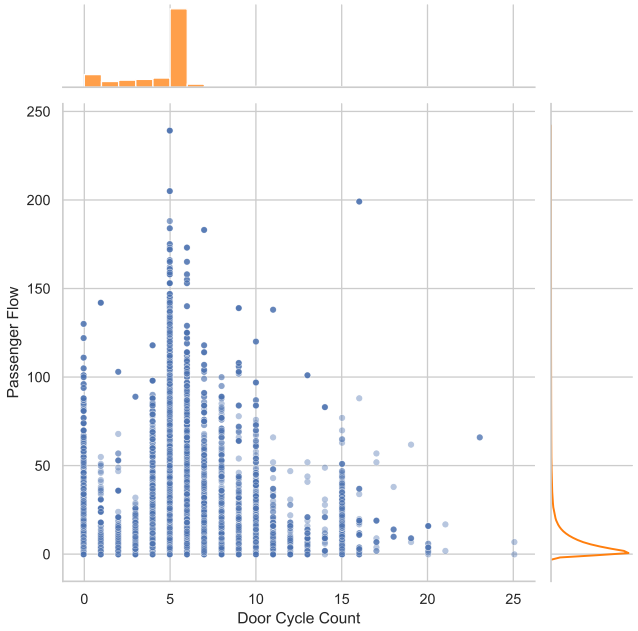

Figure 5.3: Scatter plot of door cycle counts versus passenger flow including marginal distributions of the two variables in orange on the sides of the figure.

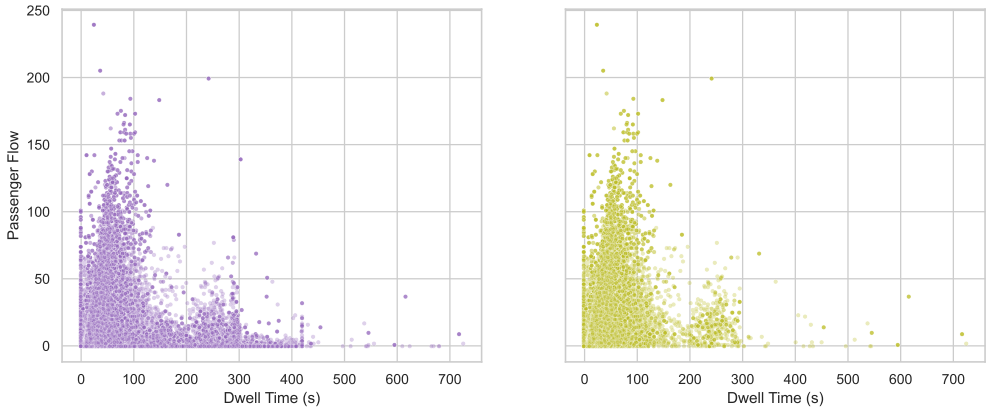

Figure 5.4: Scatter plot of dwell times versus passenger flow for all stations (left) and non-outer stations (right).

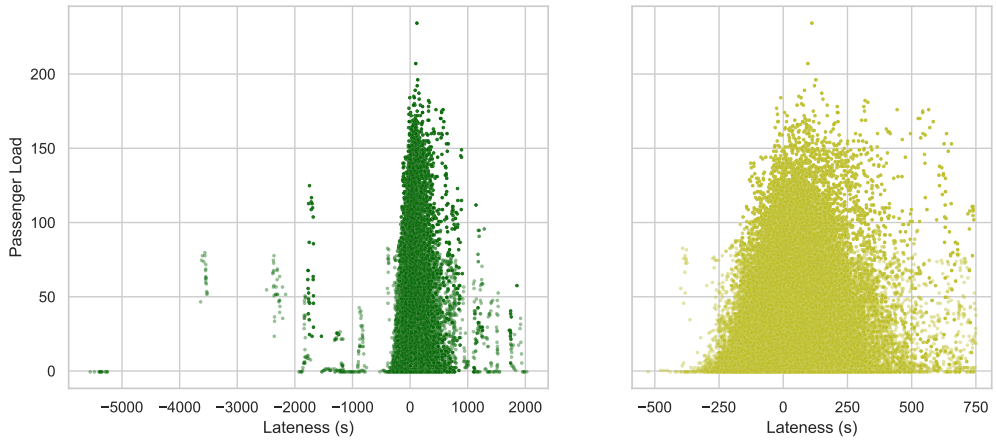

Figure 5.5: Scatter plot of vehicle lateness versus the passenger load where the left figure shows all values and the right figure shows a slice of the data with only the main region visible.

value indicates that the return air is warmer than the incoming air and a negative value indicates that the return air is colder than the incoming air. Figure 5.7 shows a scatter density plot for the temperature delta values for each individual HVAC unit. There seems to be a slight relationship between positive temperature deltas and a higher passenger load.

5.2.5. Outside Temperature

While the outside temperature is not related to the state of the vehicle itself, it is measured by the vehicle. It is measured on the link level, but the relationship to both the passenger load and flow could provide insights. Figure 5.8 shows a density scatter plot for both the relationship to the load and the flow. The distribution is similar in both cases, mostly centred around 10 and 20 degrees Celsius. While the distribution is more variant for the temperatures between 10 and 20 degrees Celsius, there does not seem to be an obvious relationship with respect to either the passenger load or flow.

5.2.6. Overall Headway

Figure 5.9 shows a scatter plot including marginal distributions of the relationship between the overall headway and the passenger load and flow. The quantitative analysis showed a slight negative correlation for the headway with respect to both the passenger load and flow. The figure seems to confirm these results. A likely reason for this is that the headway is small for “busy” stations where the required throughput of passengers is very high, such that many vehicles are scheduled to pass it. Stations in more quiet areas of the urban area may have larger headways between vehicles and thus also an overall lower passenger load. The headway, thus, is a measure of the frequency of stops in the timetable which results in such a relationship.

5.2.7. Line Number and Trip Progress

The line number was found to be of influence in the quantitative analysis for both the passenger flow and the passenger load, where some line numbers had a positive effect on the target variables, such as line number 9, and others had a negative effect, such as line number 11. Figure 5.11 where violin plots are shown with the relations for both the passenger load and flow. Line numbers 2, 9 and 15 clearly have a more long-tailed distribution towards higher passenger load and flow values. On the other hand, line numbers 11 and 17 show the opposite effect, where the distribution is centred around lower passenger loads and flow. This explains the differences in the linear relationship of the line numbers to the passenger load, where line numbers 11 and 17 had a negative correlation and line number 9 a positive correlation.

To investigate the effect that the trip progress has, in combination with the line number, on the passenger load and flow, Figure 5.11 displays the relation of the trip progress on the passenger load and flow. It is clear, again, that line numbers 2, 9 and 15 have a relatively large peak passenger load.

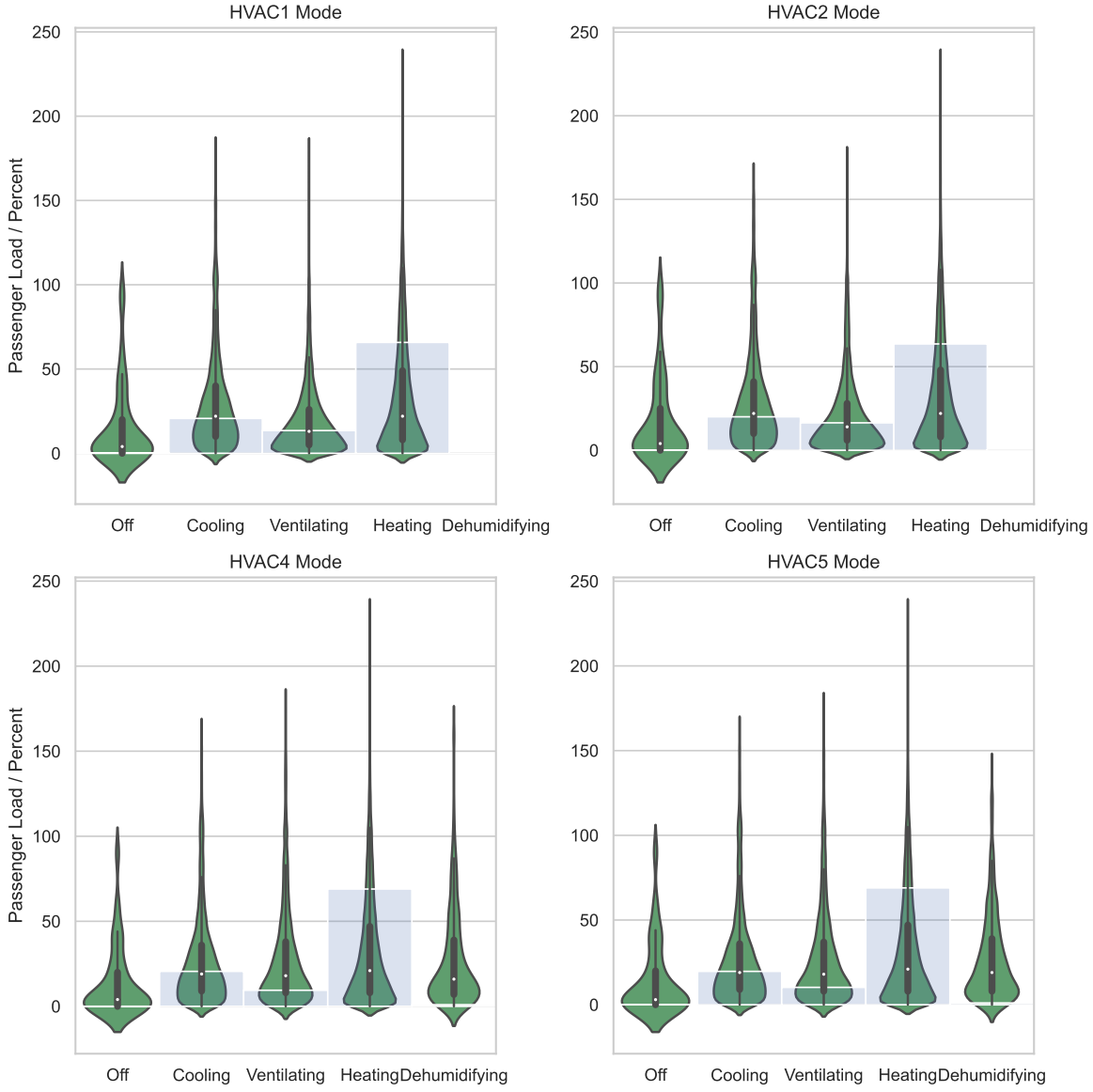

Figure 5.6: Violin plots of the different HVAC unit modes versus the passenger for each HVAC unit, the distribution of each mode is shown through the transparent histograms in percentages.

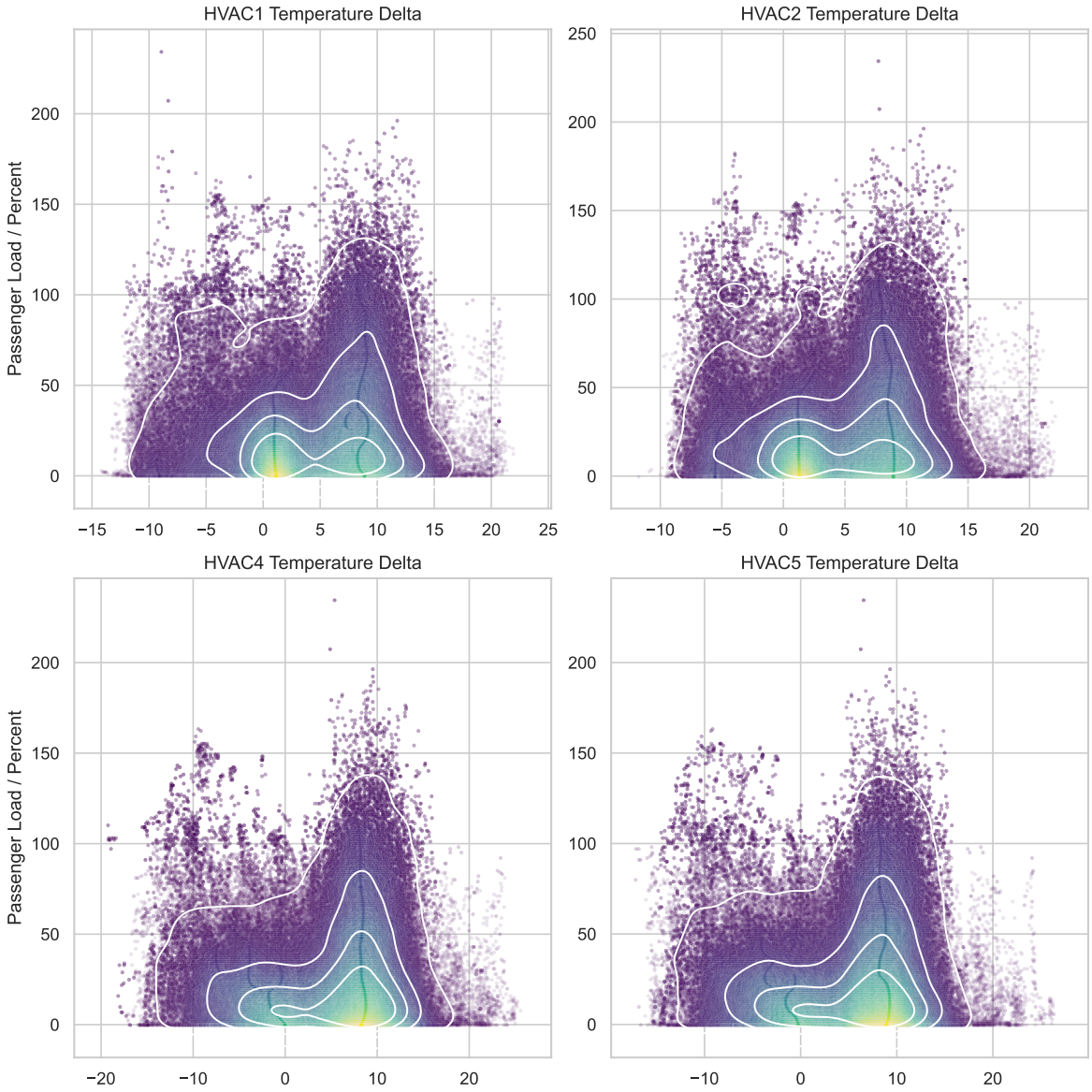

Figure 5.7: Scatter plot of the HVAC temperature delta for each HVAC unit, including colouring and regions based on estimated Gaussian kernel density.

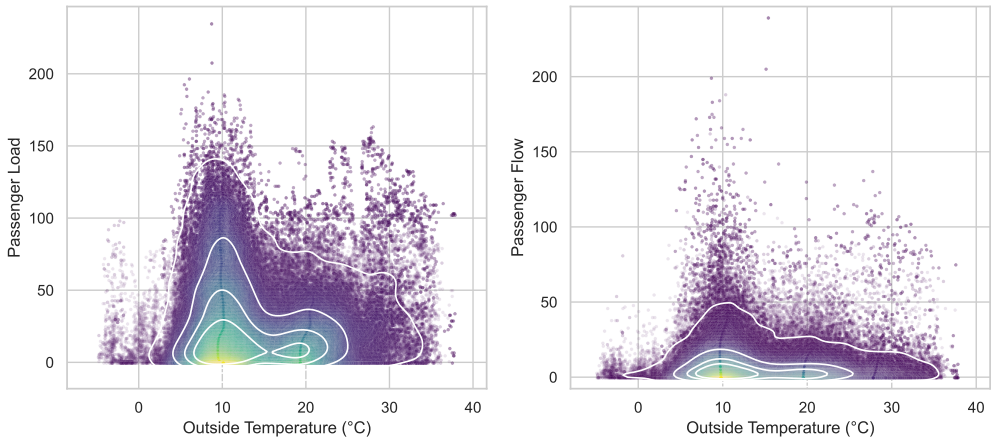

Figure 5.8: Scatter plot of the outside temperature compared to the passenger load and passenger flow including colouring and regions based on estimated Gaussian kernel density.

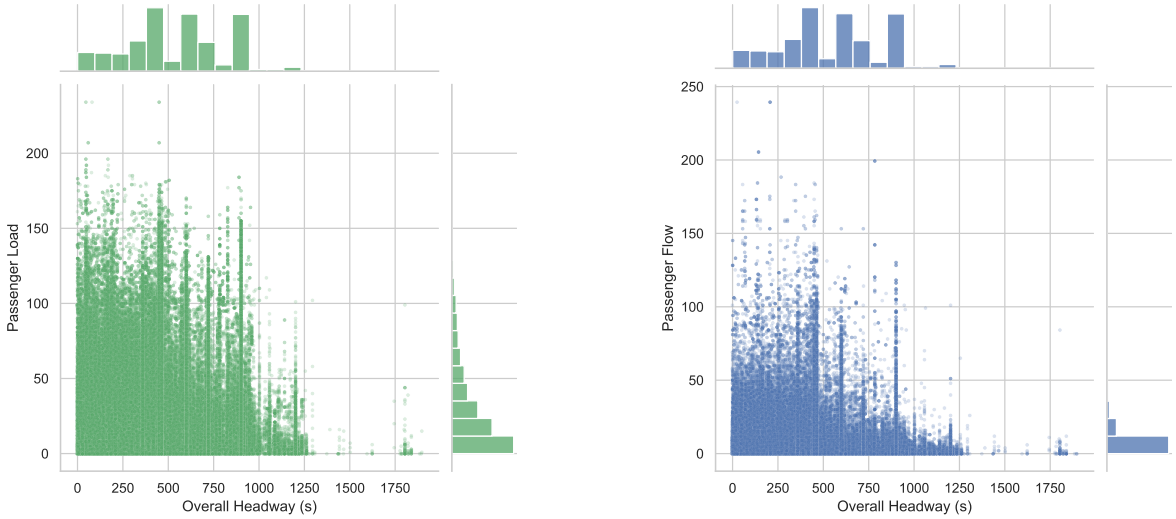

Figure 5.9: Scatter plots of the overall headway versus the passenger load (left) and the passenger flow (right) with their marginal distributions plotted on the sides of the figures.

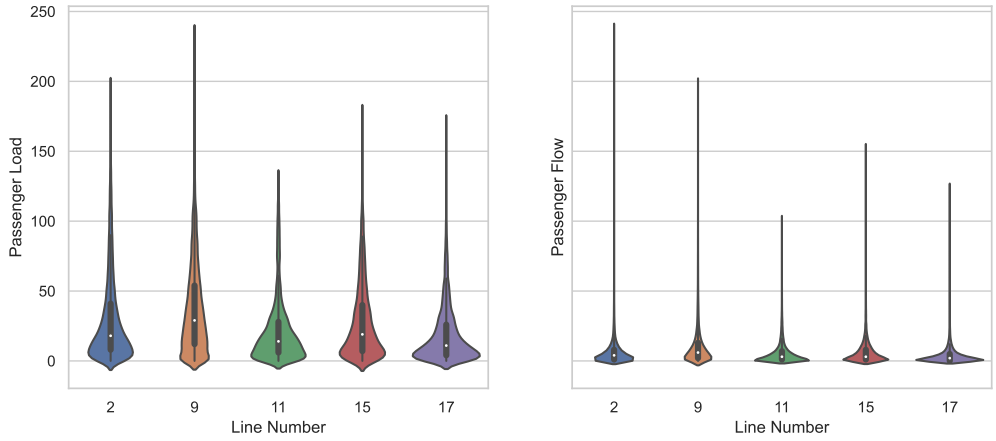

Figure 5.10: Violin plots showing the line number of the trip versus the passenger load (left) and the passenger flow (right).

It also shows that the trip progress has an effect on the values for both the passenger load and the passenger flow. For additional investigation of the passenger load with respect to the trip progress and line numbers, we refer to the exploratory AFC data analysis in Appendix A.

5.2.8. Outer Station

Finally, the ”is outer station” feature is investigated. It is demonstrated to have a negative correlation with both the passenger load and the passenger flow, see tables 5.1 and 5.2. This means that if a stop is at an outer station of the trip, the passenger load and flow should on average be lower than otherwise. Figure 5.12 shows violin plots of this relationship. The figure confirms the results of the quantitative analysis, showing very clearly that both the passenger load and flow are distributed close to zero for outer stations.

5.3. Historical AFC Features

The same feature analysis as in sections 5.1 and 5.2 has been conducted on the historical AFC features. As defined in Section 4.3, the historical AFC features are features measuring the average passenger load/flow values aggregated over similar trips in the dataset. Recall that each historical feature is defined on three temporal scales: daily, weekly and monthly. In addition, the passenger flow is modelled into two components: the board and alight count. Again, both a quantitative and a qualitative analysis are performed. However, the load features are only analysed with respect to the passenger load and the flow features in a similar fashion only to the passenger flow.

Table 5.5 shows the results of the quantitative feature analysis for the historical passenger load features, both in terms of Pearson’s correlation and the F-test as well as Mutual Information. Similarly, Table 5.6 shows the same information for the historical passenger flow features. Interestingly, all features have a strong correlation with their respective target variables ($|r| > 0.5$). For both tables, the ordering based on F value is different than based on the Mutual Information value. For the passenger load, it appears that in terms of correlation the monthly average load is the most informative feature, followed by the weekly and daily average load features. However, when considering Mutual Information, it seems that the weekly average load is the most informative feature, followed by the daily and monthly average load features.

The historical passenger flow features exhibit a similar relationship. Note that the overall strength of the relationships is weaker. This is most likely due to the nature of the relationships. Whereas the historical passenger load features are direct predictors of the passenger load, the historical passenger flow features each predicts a component of the passenger flow: the board and alight counts. Hence, the relationship between these features and the passenger flow is prone to be impacted.

A different ordering of the features based on the temporal level is apparent than for the historical passenger flow features. For both the historical board count and alight count features, the ordering

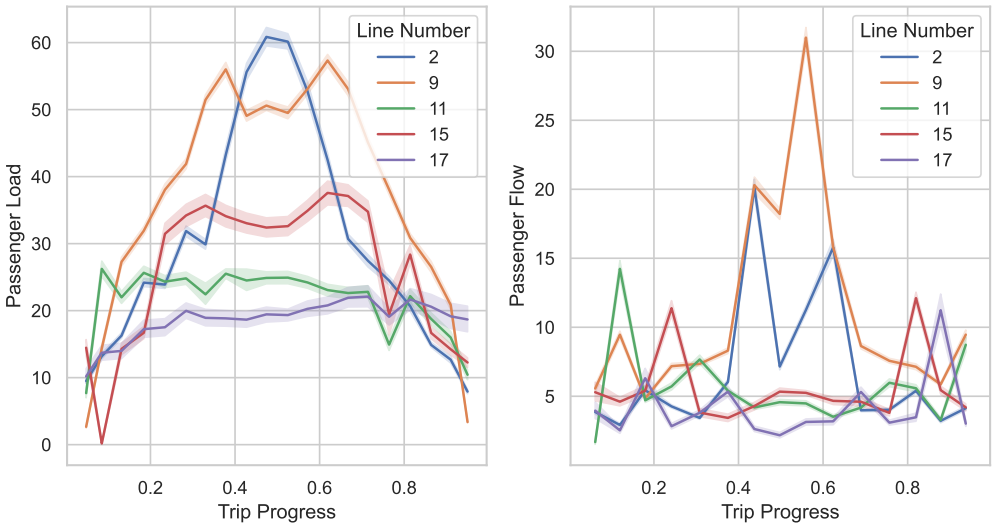

Figure 5.11: Line plots showing the relationship between the trip progress and the average passenger load (left) and the average flow (right) per line number. The trip progress has been binned to produce a more smooth figure. The plots include a 95% Confidence Interval band.

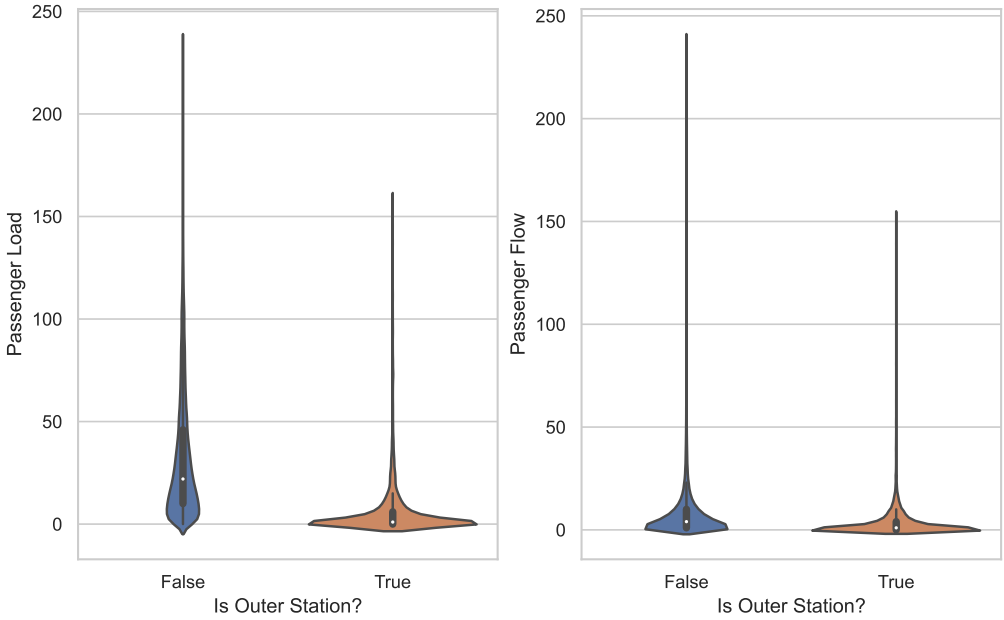

Figure 5.12: Violin plots showing the relationship between whether a stop is an outer station and the passenger load (left) and the passenger flow (right).

Feature Name	r value	F value	p value	MI value
average_monthly_load	0.77	419,857	0.0	1.10
average_weekly_load	0.76	406,815	0.0	1.42
average_daily_load	0.72	318,217	0.0	1.36

Table 5.5: Quantitative feature analysis of historical AFC features with relation to the passenger load. The r value indicates the Pearson's correlation coefficient, the F value denotes the F statistic, the p value indicates the resulting value of the F test and the MI value denotes the mutual information score. The features are sorted by the F value.

with respect to the F value is the same: weekly, monthly and daily. In terms of Mutual Information, the ordering is also the same for both feature types: daily, weekly and monthly. Observe that these orderings for the different temporal scales are thus different from the passenger load values. However, the difference in F value between the features is substantially smaller compared to the load values. Interestingly, there is almost no difference in the Mutual Information value for either the board or alight features for the same temporal level. This indicates that the board and alight counts are both equally predictive of the passenger flow in that regard.

To further investigate the nature of the relationships, figures 5.13 and 5.14 have been constructed to show the relationship of each of the historical features to its respective target variable. Observe both figures are most dense around the zero point, which is where the most frequent passenger load and flow values occur. In addition to the scatter density plot for the historical passenger load features in Figure 5.13, a linear line is plotted. If the historical passenger load features would perfectly predict the passenger load, then the data would be plotted exactly upon that line. It is immediately obvious that this is not the case. The data seems to follow a much more gentle slope. For larger passenger load values, the features seem to underestimate the passenger load significantly. The reason for this is that a high passenger load is a rare occurrence. When aggregating historical data for similar trips in such cases, the overall average will be more close to regular passenger load values. This even holds for peak moments of the day. It is therefore natural that these features would underpredict. The same will undoubtedly hold for the historical passenger flow features, while this trend is not directly obvious from the figure. Observe that while the quantitative analysis showed substantial differences in the strength of the relationships based on the temporal level of the feature, this is not so apparent from the figure. It seems that the monthly average load feature seems to have a slightly larger variance than the other features, but significant differences in the trend are not apparent.

Figure 5.14 shows the relationships between the historical passenger board and alight count features and the actual passenger flow. As mentioned previously, the relationship is likely less strong due to the features predicting only a component of the passenger flow. In all plots in the figure, there seems to be a loose linear relationship between the features and the passenger flow. However, also note the significant variance that the features exhibit. This would also explain the overall weakness of the relationship compared to the strength of the relationships of the historical passenger load features. Again, there does not seem to be a significant difference between the trend between the different temporal levels of the features. The daily historical features seem to have a lower variance than the features of the other two temporal levels.

Overall, we may conclude that the historical features are significantly related to the target variables but that they are by no means perfect predictors. Moreover, it seems that for rare instances, they increasingly lose their predictive power. The historical load features seem to underestimate the passenger load for instances with relatively high passenger loads. While based on the metrics it seems that there are slight differences between the temporal level at which the feature is constructed, the visual inspection showed that the differences were minor. Nevertheless, it seems that these historical features may contribute significantly to an accurate estimation of the passenger load overall.

5.4. Discussion

In this chapter, we have provided both a quantitative and a qualitative assessment of the relationships of the dataset features with respect to the passenger load and passenger flow.

The results show that the only vehicle-related feature that exhibits a clear relationship is the weight estimate, which seems to exhibit a linear relation to the passenger load. For the other features, the

Feature Name	r value	F value	p value	MI value
average_weekly_board_count	0.68	246,864	0.0	0.68
average_monthly_board_count	0.67	240,627	0.0	0.42
average_weekly_alight_count	-0.67	231,639	0.0	0.68
average_monthly_alight_count	-0.66	230,751	0.0	0.43
average_daily_board_count	0.65	211,636	0.0	0.77
average_daily_alight_count	-0.65	208,313	0.0	0.77

Table 5.6: Quantitative feature analysis of historical AFC features with relation to the passenger flow. The r value indicates the Pearson's correlation coefficient, the F value denotes the F statistic, the p value denotes the F test and the MI value denotes the mutual information score. The features are sorted by the F value.

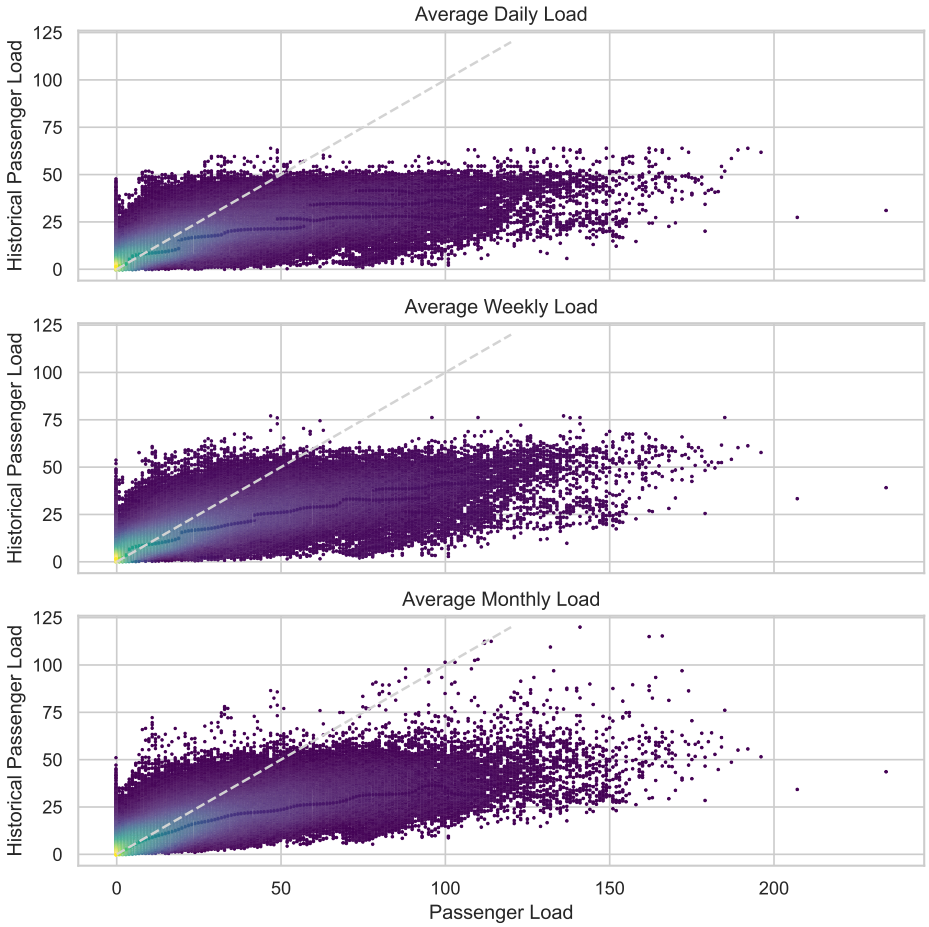

Figure 5.13: Scatter and density plot of historical passenger load features versus the actual passenger load. Each figure shows the feature on a different temporal scale: daily, weekly or monthly. A linear line is plotted in grey compare the trend of each plot to.

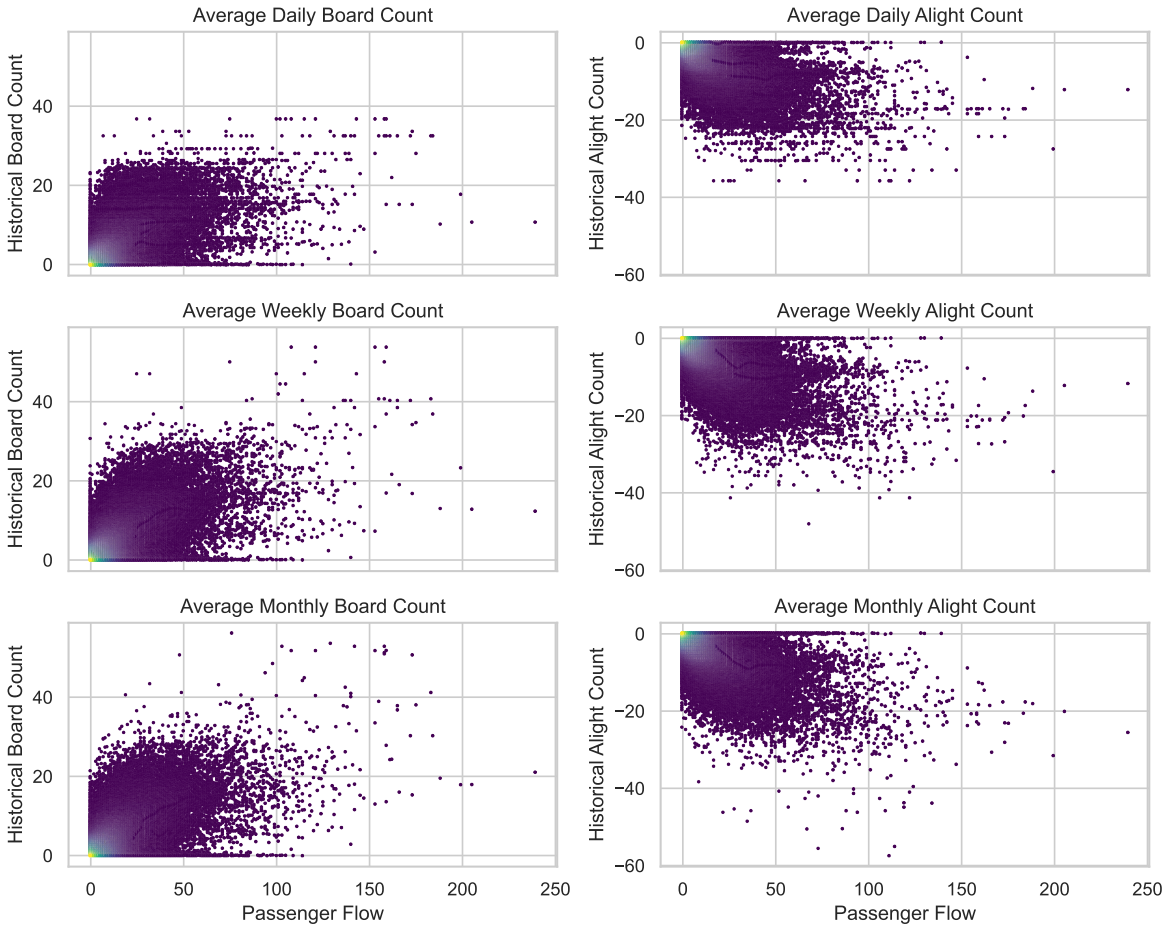

Figure 5.14: Scatter and density plot of historical passenger flow features versus the actual passenger flow. Each figure shows the feature on a different temporal scale (daily, weekly or monthly) and a different passenger flow dimension (board count and alight count). Note that alight count values are negative values as they represent the difference in passengers.

relationships are less obvious. However, the quantitative analysis showed that many features appear to have some relationship nonetheless. The HVAC-related features exhibit strong relationships as well. The HVAC mode features were linearly related to the passenger load and the temperature deltas to both the passenger load and flow. Interestingly, the temperature offset does not seem to have any strong effect whatsoever. The temperature showed a relatively strong relationship to both target variables in terms of the Mutual Information score. As expected, the door-related features had a strong relationship to the passenger flow but some of the features such as the door cycle count and the door open time had a weak relationship to the passenger load as well. The overall headway seemed more strongly related to the two target variables than the line headway.

Based on the quantitative analysis, it is clear that no single feature predicts neither the passenger load nor the passenger flow sufficiently accurate. Moreover, it appears that for estimating the passenger load a linear model would not suffice, as there are strong non-linear relationships as well. Hence, a suitable model will need to leverage both linear relationships as well as non-linear relationships. These factors should be taken into account when designing a model that incorporates these features.

This model may be complemented by the historical passenger load and flow features. These features have been demonstrated to exhibit a strong relationship to their respective target variables. In all cases, the relationship of these features is stronger than the relationship of the vehicle-related features over all metrics. Hence, these may be crucial for an accurate prediction of the passenger load. However, the historical features lose predictive power for rare instances, as they are based on historical averages and thus on “common” instances.

6

Model Design and Selection

Having conducted a feature analysis in the previous chapter, we will now describe the model design and selection. Separate models are proposed for a real-time setting and a forecasting setting. Two real-time models have been designed, one “naive” model and another using the insights derived from the feature analysis. Furthermore, we describe how the specific models have been selected and optimised. The forecasting model is based on Seasonal ARIMA but is presented with several variations such as the addition of a GARCH model or exogenous variables as described in Section 3.4.2.

6.1. Real-Time Models

The model described in this section is meant to be deployed to make real-time predictions based on incoming data. However, it can also be suitable for historical predictions given data in the appropriate format. Two models have been developed, the Passenger Load Prediction (PLP) model and the Load-Flow Fusion model (LFF) model. Both models are evaluated in Chapter 7. In this section, the specific design and selection process for each model is described.

6.1.1. Passenger Load Prediction Model

The design principle behind the PLP model is that it makes predictions based on tabular data. It is conceptually a simple model, where the features serve as input and the output is a passenger load prediction. We refer to it as a naive model as it makes no assumptions about the data that it is given or the problem at hand. The challenging aspect of this model is to find an appropriate machine learning technique for the model to use that is able to leverage the specific problem characteristics and relationships between the features and the output as discussed in Chapter 5.

A host of models have been considered as the base model for the PLP model design, ranging from linear methods to artificial neural networks. In order to get a good idea of which type of model is optimal for this scenario, an experiment has been set up where each model is trained and evaluated on a portion of the dataset, using common and sensible hyperparameters. The models are evaluated using the methods described in Section 3.6.1. The best model is selected as the base model for the PLP model design.

The subset of the dataset consists of one-third of the unique trips in the dataset and has been divided into a train set and test set with proportions 0.7 and 0.3, respectively. The entire subset has a size of 97,206 rows and 3,035 unique trips which exclude duplicated trips due to oversampling. Each model is constructed and evaluated using scikit-learn [41] using the default parameters unless otherwise stated. For the gradient boosting models, varying values for the maximum tree depth have been used. For the random forest models, varying values for the number of trees have been used. For the neural network models, varying values for the hidden layer sizes have been used. These values are indicated in parentheses. The reason that for these models different hyperparameters have been used, is that the performance is expected to be heavily influenced by these specific parameters. The results of the evaluation are shown in Table 6.1.

It is clear that gradient boosting is the most accurate method for predicting the passenger load. The method outperforms the other methods over all three metrics. Therefore, gradient boosting is selected

Model	R^2	MAE	RMSE	Normalised
Gradient Boosting (7)	0.76	8.07	11.92	
Gradient Boosting (5)	0.74	8.50	12.41	
Random Forest (200)	0.74	8.36	12.53	
Random Forest (150)	0.74	8.38	12.56	
Random Forest (100)	0.73	8.39	12.57	
Neural Network $(25, 5)$	0.71	8.83	13.21	✓
Gradient Boosting (3)	0.70	9.14	13.31	
Neural Network (100)	0.70	9.68	13.36	✓
Least Squares Normalised	0.62	10.74	15.11	✓
Least Squares	0.62	10.75	15.12	
LASSO	0.61	10.69	15.15	
Support Vector Machine	0.59	10.29	15.66	✓
LASSO Normalised	0.58	11.15	15.74	✓
Decision Tree	0.46	11.82	18.00	
k-Nearest Neighbours	0.42	12.51	18.64	✓
AdaBoost Normalised	0.16	19.69	22.38	✓
AdaBoost	0.16	19.69	22.38	

Table 6.1: Evaluation results of fitting various models to a subset of the evaluation dataset, sorted by the R^2 score. The normalised column indicates whether the feature values have been normalised around zero before fitting. The gradient boosting, random forest and neural network models are appended by additional parameters. For the gradient boosting model, this is the maximum tree depth. For the random forest model, this is the number of estimators. For the neural network model, these are the hidden layer sizes.

as the base model for the PLP model. The exact parameters and input features of the PLP model can be found in Appendix E.

6.1.2. Load-Flow Fusion Model

As we have previously demonstrated, the PLP model provides a suitable model design in the current context. However, it is in a certain way a naive approach to the problem. The LFF model is designed such that it may leverage some properties of the problem in the current scenario. These are as follows:

One obvious characteristic of the problem is that it is a time-series problem. Consider in the current scenario a vehicle that is executing a trip at a time step t, the passenger load is affected by the passenger load at time step $t-1$ and by the net passenger flow at the current stop. Typically, the passenger load starts from a low number and rises to one or several peaks and then descends towards a low number at the end again[1]. Hence, a model that is aware of its previous estimates may be able to leverage this knowledge for more accurate predictions.

Another observation of note follows from the feature analysis in Chapter 5. This is the observation that some features have a stronger relationship to the passenger flow than to the passenger load. Given the previous observation that the passenger flow affects the passenger load at the current time step, it may be beneficial to predict the passenger flow independently from the passenger load using the related features respectively.

The LFF model design incorporates these two observations and thus proposes an architecture consisting of two sub-models, or so-called components: one is the flow component and the other is the load component. The flow component predicts the passenger flow at the current stop based on the flow-related features, the load component predicts the passenger load at the current stop using the load-related features. The prediction of the flow component is added to the previous time step's passenger load prediction to get a new passenger load prediction. Note that if the current stop is the first stop of a trip, the previous estimate is taken to be zero[2]. The load component is similar to the PLP model described in the previous section in that it directly predicts the passenger load at each time step

[1] Again, we refer to the exploratory data analysis in Appendix A for further analysis of the passenger load patterns in the current context.

[2] As discussed in Section 4.2.3, the assumption that trips start and end with zero passengers does not always hold. The goal of the current design is that in such situations, the load component's prediction can counterbalance this inaccuracy.

without considering previous predictions.

Both passenger load predictions are then passed to a fusion component that weighs both predictions and produces a final prediction. This fusion layer incorporates additional features as well in order to weigh the predictions by the components based on the context. The intuition is that there is a trade-off between the two components. The flow component follows a time-series based approach and could thus capture the trend in the passenger load better throughout the trip. However, the consequence of this approach is that errors accumulate for each prediction. On the other hand, the load component does not accumulate errors but disregards the trend overall. Therefore, depending on the context, the fusion component may weigh the prediction of each component differently to derive a better prediction.

Before describing the technical details of each of the components, it is worth mentioning the process of fitting the LFF model as it is a non-trivial process. As the fusion component relies on the output of the load and flow components for its input, the LFF model needs to be fitted in two stages. To facilitate this, the training dataset needs to be split into two portions. First, both the load and flow components are fitted to the first portion. Then, both components make predictions on the unseen second portion which are subsequently used to fit the fusion component to. During an evaluation, both the load and flow components make their respective predictions which are passed to the fusion component which makes the final prediction. Note that in order for the flow component to be able to make passenger load predictions, all preceding stops of the trip under evaluation should be passed to the model.

Based on the results of the PLP model selection experiment in Table 6.1, the gradient boosting machine is selected as a suitable model for both the passenger load and passenger flow components.

The value for passenger flow can be produced in varying forms. The passenger flow component could, for instance, produce a net-flow value, a gross-flow value or separate values for the board and alight count. A net-flow value is the number of boarding passengers subtracted by the number of alighting passengers and a gross-flow value is the total number of boarding and alighting passengers. This flow value is used in the feature analysis of Chapter 5. Initially, the component was implemented to produce a gross-flow value which would be weighted by a flow coefficient based on historical data. This flow coefficient would be calculated as the average board count subtracted by the average alight count divided by the sum of the two average counts on similar times of days for the same trip. This produced a highly inaccurate model, as the error of the calculation introduced by the model producing the total flow was compounded by errors in the flow coefficient. As it provides the highest prediction accuracy, based on exploratory analysis, it is decided to let the component provide the board count and alight count as separate values. An additional benefit of this method is that it provides more interpretability to the model's output. As gradient boosting models are natively not able to provide multiple outputs, two models for predicting the board and alight count are trained separately. However, both these models are fitted using the same hyperparameters.

The goal of the fusion component is to intelligently aggregate the intermediate outputs of both components. To do so optimally, the aggregation needs to be based on a notion of the context such as the time of day or line number. Hence, a conditional approach based on decision trees seems appropriate for the model. Two ensemble methods have been experimented with: a random forest model and a gradient boosting machine. During exploratory analysis, the gradient boosting machine showed to best results and was therefore selected as the fusion component's model. An additional benefit of using the gradient boosting machine method for the fusion component is that it allows for model interpretation using SHAP values as described in Section 3.6.2.

The features for each of the individual components have been selected based on the strength of the relationship to either the passenger load or flow. A significant group of features is provided to both components due to them being related to both target outputs. A group of base features is provided to all components, including the fusion component. This group consists of temporal features, which are the time of day and day of the week, as well as GTFS related features, which are the line number, the trip progress and the headways. The load component is additionally provided with the weight estimate, all HVAC features, the outside temperature, the total door open time as well as the door cycle count and the historical load features. The flow component is additionally provided all the door-related features, the weight estimate, the HVAC temperature deltas, the outside temperature and the historical flow features. A complete overview of the input features for each component is given in Appendix E. Furthermore, historical AFC data features are passed to both the load and flow components as described in Section 4.3. The load component is provided historical passenger load features and the flow component is provided historical passenger flow features.

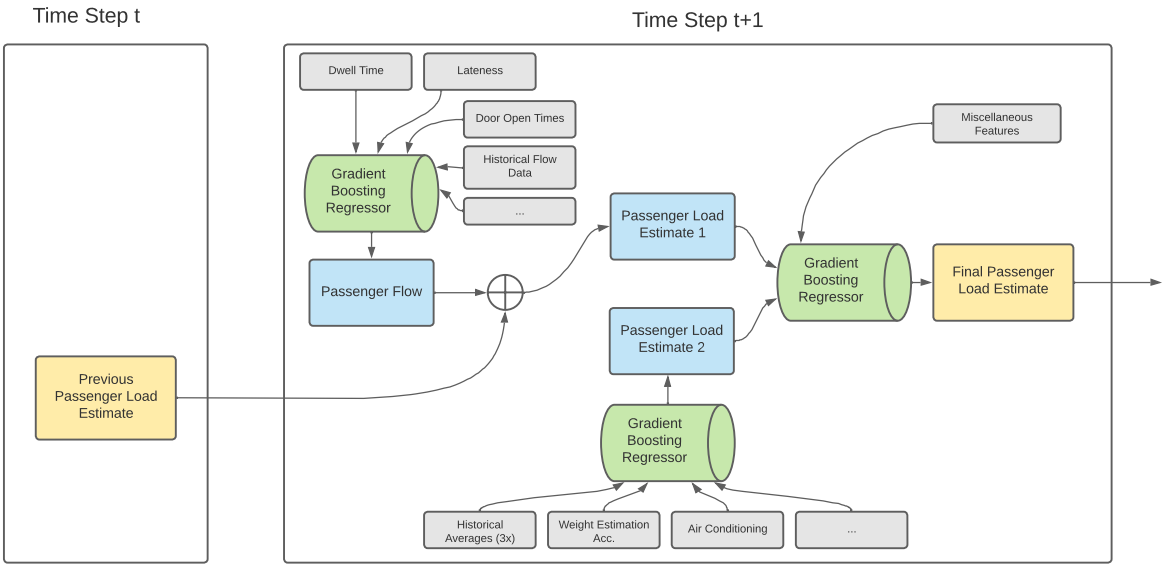

Figure 6.1: Schematic diagram of the LFF model's design showing the input features in grey, the model components in green, the intermediate estimates in blue and the final estimates in yellow.

Figure 6.1 shows a schematic diagram of the proposed LFF model design.

6.1.3. Optimisation

The model designs provided a basis for the implementation of the models. However, the models are not optimised toward the current setting. Most importantly, the hyperparameters used so far were the default arguments used by the individual models. In order to find the optimal hyperparameters for the problem scenario at hand, a grid search is set up that finds the optimal hyperparameters for each individual model. Note that the two models of the flow component are optimised jointly using the same hyperparameters. The same set of varying hyperparameters is applied to each gradient boosting model.

The hyperparameters for gradient boosting machines can be divided into two categories. The first category is boosting-related parameters which influence how the ensemble of trees grows and the second category is tree-related which influences the growth of the individual trees in the ensemble. The boosting and tree components are optimised independently in order to reduce the exponential growth of the hyperparameter sets. In order to decide which hyperparameter set is optimal, the scores of the three metrics given in Section 3.6 are compared across sets. If there is no optimal set that can be found because some set is more optimal in one of the three metrics than others, then the least complex hyperparameter set is selected that has an optimal score in at least one of the metrics.

The specific values for each hyperparameter to be tuned can be found in Appendix D, as well as the results of the grid search and the final optimal parameters.

6.2. Forecasting Model

In addition to the real-time models, we also propose a forecasting model design. This model should be able to make short-term forecasts from a vehicle-centric approach and be able to update based on incoming signals. The goal of this section is to select a model based on the existing literature and describe how real-time information can be used to update the model. Several variations on the model are presented and all variations are evaluated in Chapter 7.

Based on the related work overview in Section 2.1, it is evident that forecasting methods predominantly use one of four methods: ARIMA, SVR, Kalman Filtering or LSTM. It is important that the model can make dynamic forecasts and can update itself using real-time data.

Out of the four models, it seems that ARIMA is the most suitable in the current scenario. A model such as SVR is not designed to update frequently on incoming signals. Kalman filtering does not allow

modelling using features such as historical data. While the LSTM model is a suitable candidate as well, the complexity and size of the Deep Learning model limit its applicability in the current context.

Due to the flexibility of the model, the time-series nature of the data and the intrinsic seasonality of the passenger load data, the Seasonal ARIMA model is used as the base model for the forecasting model. This model has widely been successfully applied in the literature for passenger load and flow forecasting [6, 20, 21, 23, 25].

Variations of Seasonal ARIMA exist, where additional factors can be modelled. These will be evaluated as well. The variations under consideration are: modelling volatility using GARCH and performing regression with Seasonal ARIMA errors. In addition, incorporating both of these variations into a single model is considered as well.

Note that the model can be applied to only a single time series due to the differences in characteristics between different time series. Here, a single time series represents the trips over a single route. Thus, if the passenger load is to be estimated for the remainder of a vehicle's trip, an ARIMA model is constructed for the trip using a *history* of similar trips on the same route that occurred before.

6.2.1. Seasonal ARIMA

One of the identified research gaps in Section 2.1.3 is that the literature does not evaluate how the prediction accuracy dynamically changes throughout the forecasting horizon by observing incoming passenger load values during a trip. This will be an additional research objective of the forecasting model and the Seasonal ARIMA model will thus be used to update as well.

The order parameters for the model are selected using an automated method as implemented by the pmdarima library [69] which applies a grid search while optimising for the AIC score.

6.2.2. Model Variations

Section 2.4.3 has introduced the GARCH model and its fusion with the Seasonal ARIMA model. For the current forecasting model, GARCH is used to model the volatility of passenger loads by fitting to the residuals of Seasonal ARIMA.

A similar fusion of GARCH with ARIMA has been performed by Ding et al. [20] which showed promising results, namely that GARCH is able to effectively model the stochastic volatility in passenger load patterns and that ARIMA is effective in modelling the deterministic mean passenger load. Therefore these models may complement each other. On the other hand, related work by Chen et al. [26] has found that GARCH does not contribute to a better prediction for Seasonal ARIMA and may even deteriorate performance. It will thus be interesting to analyse its effects on the model.

For this experiment, the GARCH (1,1) model is used as it is the most simple and robust method [67].

In order to use some of the available features based on the real-time data in the forecasting model, exogenous variables are introduced into the model to create a regression model with SARIMA errors as described in Section 2.4.2. The challenge is to find features such that they do not violate the constraints required for the model. Thus, they need to be linearly related, stationary in the same order and need to be defined over the entire forecasting horizon.

None of the real-time vehicle features can be used due to them not being available over the forecasting horizon. That leaves the historical, temporal and GTFS features as candidates for exogenous variables. The historical features are expected to be linearly related to as well as stationary in the same order of the passenger load as they are derived values of the historical passenger load, this has also been demonstrated in Section 5.3. Based on the stationarity of the values as well as their linear relation to the passenger load, the historical load features on a daily, weekly and monthly basis are selected as exogenous variables. The stationarity of the features is confirmed by an Augmented Dickey-Fuller (ADF) test [70], for which the test statistics were all significant. None of the GTFS features achieved a significant p value for the ADF test, therefore these are not used as exogenous variables in the model. Figure 6.2 shows the relationship in a time series sense between these features and the passenger load.

Finally, to test all variations jointly, a final model variation is constructed combining regression with SARIMA errors and GARCH. This poses a challenge, as SARIMA and GARCH would both be modelling errors. To address this the model is constructed as follows:

$$y_t = \beta_t x_t + u_t + \epsilon_t, \qquad \epsilon_t \sim N(0, \sigma_t^2) \tag{6.1}$$

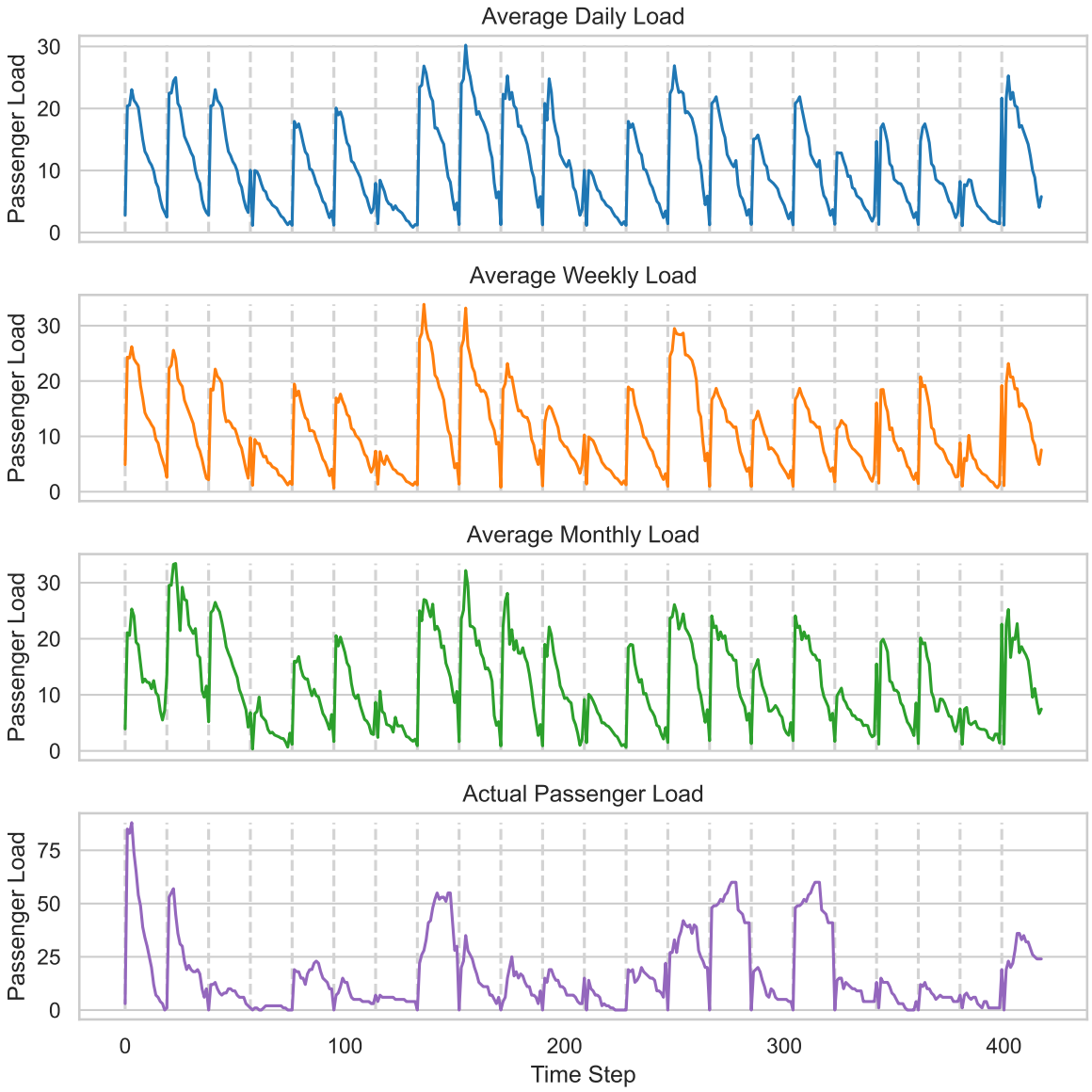

Figure 6.2: Visualisation on a subset of the data on line 11 from Rijswijkseplein to Scheveningen Haven showing values for the historical load features and the actual passenger load. The figure indicates the stationarity of the different time series where the dashed vertical lines indicate the start of a new trip.

where $\beta_t x_t$ is the exogenous regression part, u_t denote the SARIMA errors as per Equation (2.12) and ϵ_t denote the GARCH error as per Equation (2.13).

6.2.3. Real-time Signals

The main objective of the forecasting model is to be able to update itself using incoming real-time data from the vehicle. As described previously, vehicle-related features as real-time signals cannot be used as exogenous variables. However, one could use the predictions from one of the real-time models as proposed in Section 6.1, which is based on the real-time sensor data, to update the model. In that way, the forecasting model is indirectly updated using information from the real-time data, albeit using the real-time model as an intermediary. Note that the errors in forecasting are compounded by the error of the real-time model. Hence, it is critical that the real-time model has a reasonable error score for the forecasting model to be able to make accurate forecasts.

6.3. Discussion

In this chapter, we have presented the model designs for both a real-time setting as well as a forecasting setting for the passenger load prediction problem in the current context.

Two real-time models have been presented: the PLP and the LFF model. The PLP model could be considered to be a naive model, not leveraging any specific structure in the problem or of the features. It relies on the gradient boosting model's ability to extract information from the provided training data. On the other hand, we have provided a design of the LFF model, which takes into account several observations from both the problem structure and the feature analysis. It consists of several components, each relying on a different set of features. It also performs a time-series method of prediction, following the structure of the problem. While each individual component may suffer from weaknesses, such as the accumulation of errors, the fusion component is intended to act as a balancer which intelligently combines the predictions of the two components.

The forecasting model design follows the state-of-the-art in the literature, it is a Seasonal ARIMA model. However, we presented several variations of the model to evaluate as well. The first variation is adding GARCH to the model. This has been studied in the literature with mixed results. The second variation is performing regression with Seasonal ARIMA errors, meaning that exogenous variables are modelled linearly with a variance based on Seasonal ARIMA. The variables used are the historical passenger load features as they are linearly related to the actual passenger load and are stationary in the same order. A limitation of the current design of the forecasting model is that the features from the vehicle are not directly incorporated into the Seasonal ARIMA model. This is not possible due to the restrictions on these features as outlined in Section 6.2.2. Finally, the GARCH and regression with Seasonal ARIMA errors variations are combined. The evaluation will indicate which of these variations is optimal. Note that in the evaluation, a real-time experiment is simulated where, as the trip progresses, incoming real-time information from the vehicle can be used to update the model. This was a notable research gap in the literature and an interesting research objective. By investigating this, the contribution of real-time signals to a forecasting model can be quantified. In this way, the model serves as an intermediary toward the features extracted from the vehicle.

7

Results and Discussion

In this chapter, we present the experimental evaluation results of both the real-time models as well as the forecasting models. Using a SHAP analysis, the most accurate real-time model is interpreted and the results are compared to the feature analysis from Chapter 5. To investigate the contribution of the vehicle-related features, we perform an ablation study on the LFF model. Finally, the forecasting model evaluation does not only quantify the performance of each model but also how incoming real-time signals of the passenger load can help improve the performance. Each section of results is concluded by a summary and discussion of the presented results.

7.1. Real-Time Models

Section 6.1 presented two real-time models: the PLP and LFF model. The models are evaluated on a dataset through three different metrics. A baseline model is evaluated as well to serve as a comparison to the literature.

7.1.1. Baseline

It is important to compare the results of the models to a suitable baseline that is similar to the state of the art in related work. The comparison of the results to the baseline allows one to draw a conclusion about the general performance of the models.

The random forest model has been applied in various works related to passenger flow prediction [17, 35, 36, 51] and has proven to be an effective method for predicting ridership in public transport vehicles, in particular when compared to DL approaches. As a suitable baseline for this experiment, a random forest model resembling the calendar model as presented by Toqué et al. [71] is used. The input features are all the historical passenger load features as well as all non-real-time features originating from the AFC and GTFS data.

Similar to the real-time models, the baseline is also optimised using a grid search applying the procedure described in Section 6.1.3. The results of the optimisation and specific information regarding the details of the model can be found in appendices D and E.

7.1.2. Experimental Setup

A dataset has been collected on which the two real-time models and the baseline will be trained and evaluated. The data engineering pipeline described in Chapter 4 is used to collect this dataset. For the experiment, instances from 15,000 trips have been selected. As the data is highly unbalanced towards trips with an average number of passengers at its peak, balancing by means of oversampling has been performed.

The procedure of oversampling is as follows. Eight bins are created between 0 and 120[1] and another bin from 120 to ∞. Then, all trips in the AFC collection are grouped by their peak passenger load. For each bin $15000/9 \approx 1666$ trips are sampled with a peak load in that bin. If more trips need to

[1] This cutoff at 120 is selected since roughly only 0.5% of all of the trips have a peak load of more than 120. In fact, approximately 80% of trips have a peak load of less than 40.

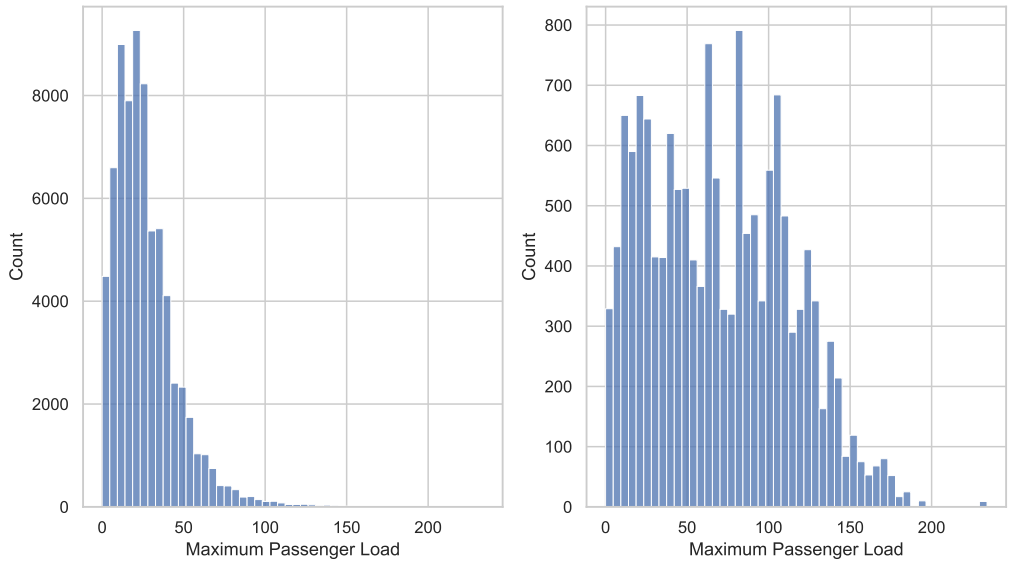

Figure 7.1: Distribution of the peak passenger load in trips in the whole AFC collection (left) and the oversampled dataset used in the evaluation (right).

be oversampled trips are removed from the test set before evaluation to ensure that no duplicate trips are present in the test set. Figure 7.1 displays the imbalance of the passenger load in the dataset and the result of oversampling[2].

The result of the sampling procedure yielded a dataset of 15,000 trips containing 9,114 unique trips. Trips have been oversampled up to 11 times. Due to errors in the data collection process, the final dataset contains 12,204 trips with 291,486 rows in total.

The dataset is split into three portions of relative sizes 0.6, 0.2 and 0.2, respectively. The first portion is used for fitting the PLP model, baseline and the load and flow components of the LFF model, the second is used for optimisation and fitting the fusion component and the final set is used for evaluation. To ensure that results are consistent, all results are cross-validated five times. The results will show the mean scores of the cross-validation.

For the LFF model, it is important to analyse the performance of the two underlying components and thus the evaluation results for their intermediate predictions on the test set are given as well. The performance of both the load and flow components are given for the same three metrics and compared to the improvements made by the fusion model.

As described in Section 3.6.1, the real-time models are also evaluated with respect to their performance in providing crowding indicators. For this evaluation, the models are not cross-validated and a single data split of the dataset is used.

Needless to say, when predicting crowding indicators, the number of bins used and the specific strategy for binning have a large influence on the perceived performance of the models. The binning strategy used here is the uniform binning strategy, which creates equally sized bins over the range of values. While various strategies may be applied, the uniform strategy is a simple and intuitive method of modelling crowding indicators. Note that the bin sizes are created based on the collection of both the predictions and ground-truth labels to avoid values falling outside of the bins. The amount of bins is set to six. Related work seems to focus on three bins [9, 37] or five bins [6] and travel planners also usually show three bins (see Figure 3.1), but with six bins there is a larger degree of granularity which allows for a better comparison of model accuracy. The consequent bin width is roughly 30.

[2]Further analysis of the distribution of the passenger load and other characteristics of the AFC data is provided in the exploratory data analysis in Appendix A.

	LFF		PLP		Baseline	
	Mean	SD	Mean	SD	Mean	SD
RMSE	*11.60*	0.189	11.63	0.302	18.39	0.560
MAE	*7.83*	0.082	7.87	0.141	12.49	0.302
$\mathbf{R^2}$	*0.844*	0.012	*0.844*	0.009	0.610	0.003

Table 7.1: Experimental results of evaluating the LFF, PLP, and baseline models on the test set. All results show the mean and standard deviation (SD) metric scores of a 5-fold cross-validation. The best mean scores are written in italics.

Metric	LFF	PLP	Baseline	Dummy
Macro F1	0.490	0.464	0.387	0.138
Weighted F1	0.828	0.823	0.767	0.587

Table 7.2: Macro and weighted F1 scores for crowding indicator predictions over a 6-binned dataset by the LFF, PLP and baseline models, along with results of a dummy model that only predicts zero.

7.1.3. Evaluation Results

Table 7.1 shows the results of the previously described experiment. The metrics are computed as per equations (3.4), (3.3) and (3.5). As a 5-fold cross-validation is performed, the standard deviations of the values are provided together with the mean value for each of the metrics. The best mean score for each metric is written in italics.

Using an arbitrary fold of the cross-validation experiment, the model predictions are grouped into six bins and the classification performance for crowding indicators of the models is evaluated. The macro and weighted F1 scores for all bins are shown in Table 7.3, including the results of a dummy predictor which only outputs zero. The F1 score per bin is shown in Table 7.3, which also shows the support for each bin. Figures 7.2, 7.3 and 7.4 show the confusion matrices for the LFF, PLP and baseline models.

7.1.4. Error Analysis

To further evaluate the models, we conduct an error analysis with the purpose to find certain aspects where either of the PLP and LFF models may perform better than the other. This error analysis is split into two parts. First, an error analysis is provided comparing the LFF to the PLP model. Second, an error analysis is provided comparing the load and flow components of the LFF model to further understand the predictive behaviour of the LFF model. Recall that all analyses are conducted on the results of the models on the test set using one arbitrary test-set fold from the cross-validated experiment.

The distribution of the MAE per trip for both models is shown in Figure 7.5. From the figure, it seems that the LFF has slightly more trips with a lower MAE, left of the peak and that the PLP model has more trips with a relatively high MAE, right of the peak. This indicates that the overall difference in MAE is due to a larger number of trips with a low MAE score.

Figure 7.6 shows the relation between absolute error for increasing passenger loads. The error of both models increases for instances with larger passenger loads. Note that these instances are also increasingly less frequent in the dataset. While both models perform roughly equally for instances below 90 passengers, the LFF model has a better performance for the rare high-load instances. Figure

Bin	LFF	PLP	Baseline	Support
(0, 31)	0.919	0.917	0.875	30,565
(31, 61)	0.634	0.635	0.564	9,038
(61, 92)	0.569	0.540	0.388	2,814
(92, 123)	0.419	0.346	0.240	610
(123, 153)	0.398	0.343	0.256	141
(153, 184)	0.0	0.0	0.0	24

Table 7.3: The F1 scores of the LFF, PLP and baseline models for each passenger load bin in the 6-binned dataset. The support for each bin is provided as well.

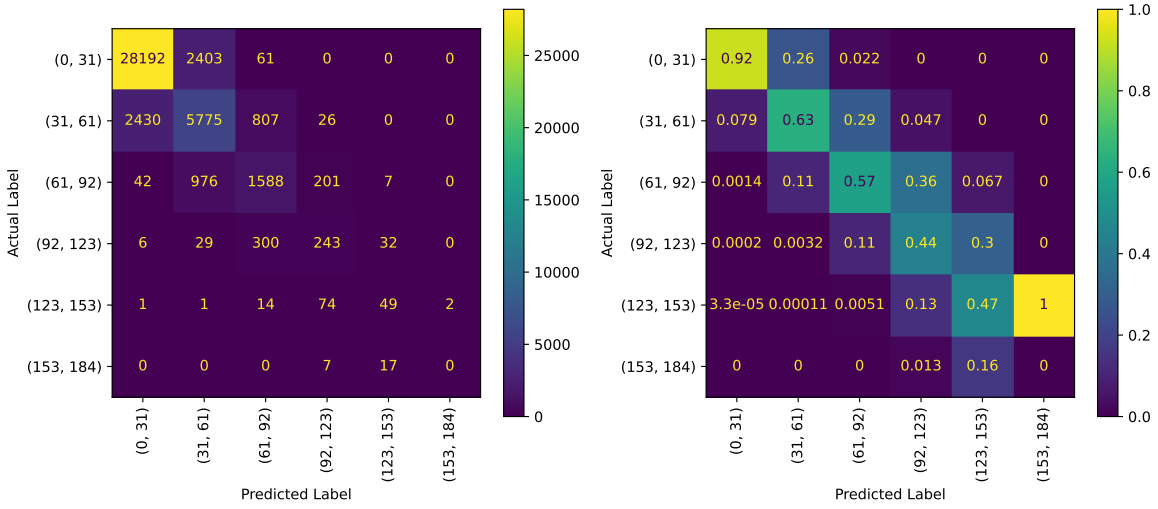

Figure 7.2: Confusion matrix of predicting crowding indicators for the LFF model showing the prediction counts (left) and the predictions normalised by the predicted label (right).

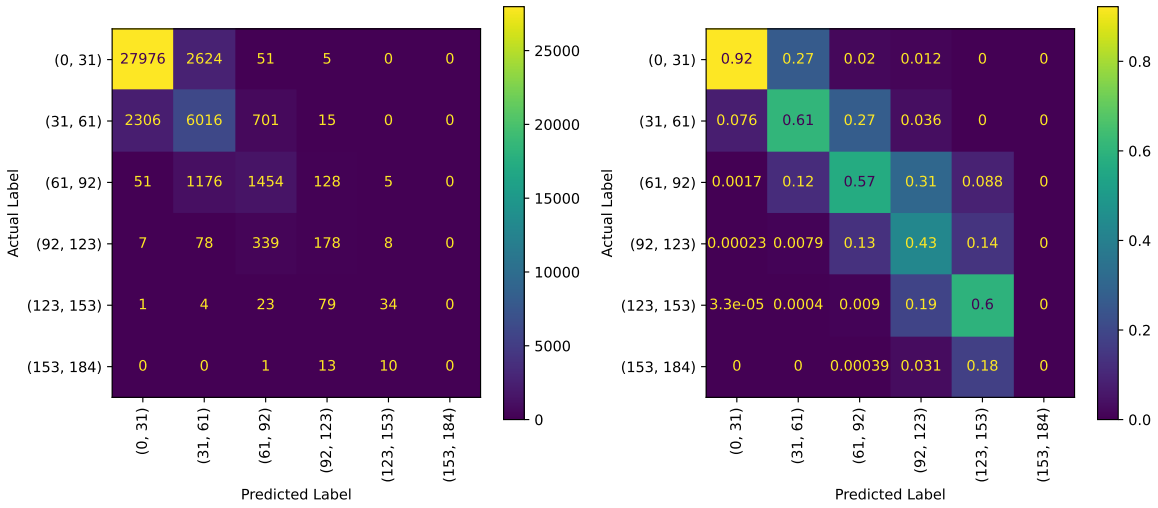

Figure 7.3: Confusion matrix of predicting crowding indicators for the PLP model showing the prediction counts (left) and the predictions normalised by the predicted label (right).

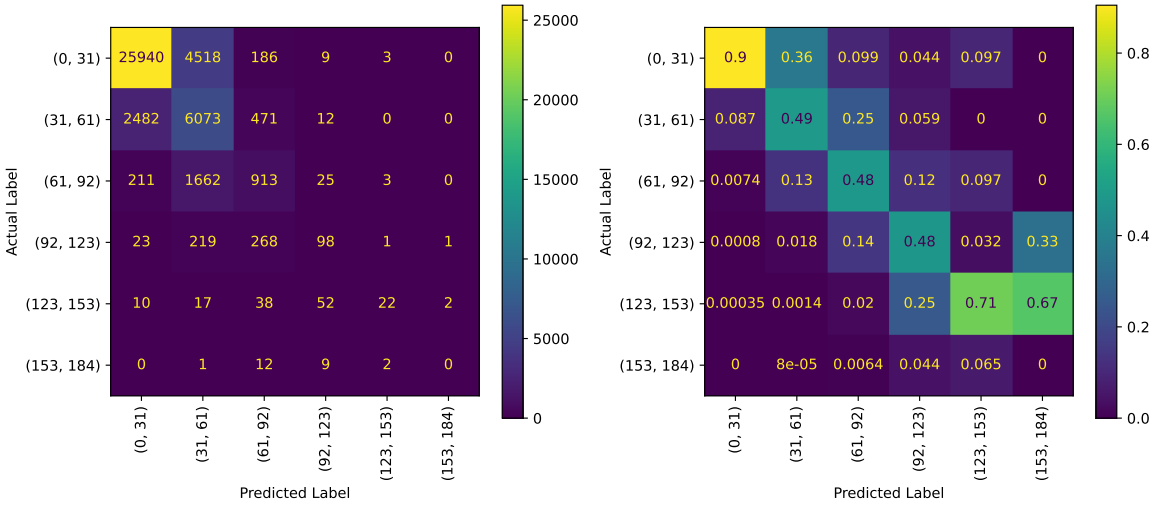

Figure 7.4: Confusion matrix of predicting crowding indicators for the test set for the baseline model showing the prediction counts (left) and the predictions normalised by the predicted label (right).

7.7 shows a similar figure, showing the errors for values of the passenger flow. Here, there seems to still be an increase in error for larger values of the passenger flow, but less significantly so. Moreover, for the highest values of the passenger flow, the LFF model performs worse than the PLP model. On the other hand, there were only a limited number of instances with flow values that high. Therefore, similarly to the instances with large passenger load values, it is difficult to draw conclusions from these results due to the small sample size. Overall, it seems that the models have a similar performance for low and common passenger load and flow values but that the LFF model has better accuracy for larger and more rare passenger load values, with the exception of several outliers.

Figure 7.8 shows the change in error over varying parts of a trip. It is clear that while the LFF model is more accurate at the beginning of a trip, it seems that the PLP model is more accurate towards the end of a trip. This could be due to the compounded error in the flow component towards the end of each trip. Once again, it seems that the performance is highly similar for both models.

We also evaluate the performance of each model and the different line numbers. Figure 7.9 shows violin plots for each line number and model and the MAE. The figure shows the distribution of MAE values calculated per trip on that line. The differences are subtle but it seems that the LFF model has an overall lower MAE for lines 2, 9 and 11. For line number 15, it seems that the LFF model has a longer-tailed distribution than the PLP model while the mean is slightly lower. For line number 17, the mean MAE of the LFF model is lower while the distribution does have a longer tail. It appears that overall the LFF model performs better but that for lines 15 and 17 the performance is more mixed.

It does not seem that any other feature of the dataset is correlated with the absolute error of either model. A Pearson's correlation analysis shows that the historical load features have a correlation to the absolute error of both models of approximately 0.4, but it is likely that it is mostly due to their correlation to the passenger load itself which is consequently highly correlated with the absolute error.

Recall that the LFF model consists of three components and predicts in two stages: first, the load and flow components predict the passenger load and then the fusion component aggregates these predictions, including some additional features, into a final output prediction. In order to evaluate the contribution of each component, the intermediate predictions of the load and flow components can be evaluated.

Table 7.4 shows the mean metric scores for the LFF model's sub-components in the same cross-validation experiment as shown in Table 7.1. The fusion component is able to improve significantly upon the individual components, as is evident from the metric scores. To further investigate this, the distribution of the difference in absolute error scores between the two sub-components is plotted in Figure 7.10 for all instances. Although the metric scores may seem to indicate otherwise, each model has a better prediction than the other model in roughly half of the cases. In fact, it appears that in 51% of the instances the load component is better and in 47% of the instances, the flow component is better. Observe, however, that while the components are competitive and equally outperform each other, the

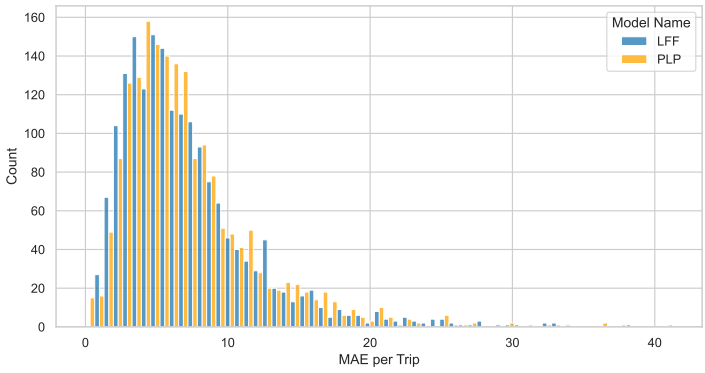

Figure 7.5: Distribution of MAE per trip in the test set for the LFF model and the PLP model.

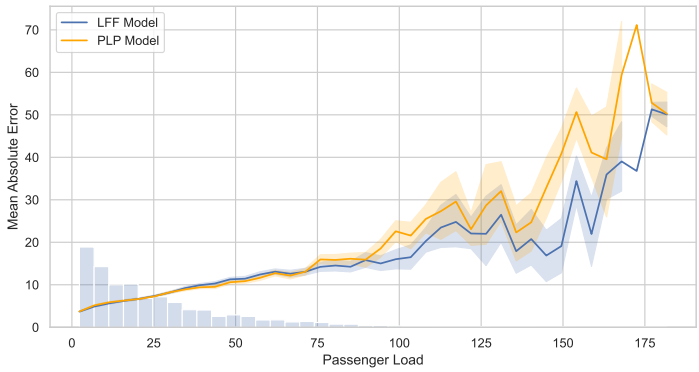

Figure 7.6: Distribution of the MAE over varying passenger loads of the LFF and PLP models in the test set. The two lines show a 95% Confidence Interval computed using bootstrapping. The passenger load values have been binned in order to smooth the curve slightly. The percentual distribution of the passenger load values is shown transparently in the figure as well.

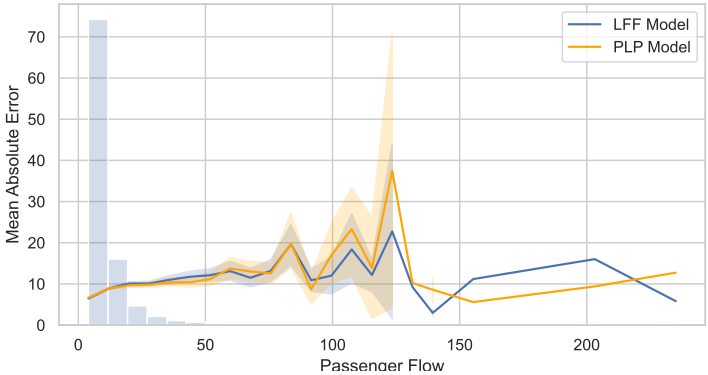

Figure 7.7: Distribution of the MAE over varying passenger flows of the LFF model and the PLP model in the test set. The two lines show a 95% Confidence Interval computed using bootstrapping. The passenger flow values have been binned in order to smooth the curve slightly. The percentual distribution of the passenger flow values is shown transparently in the figure as well.

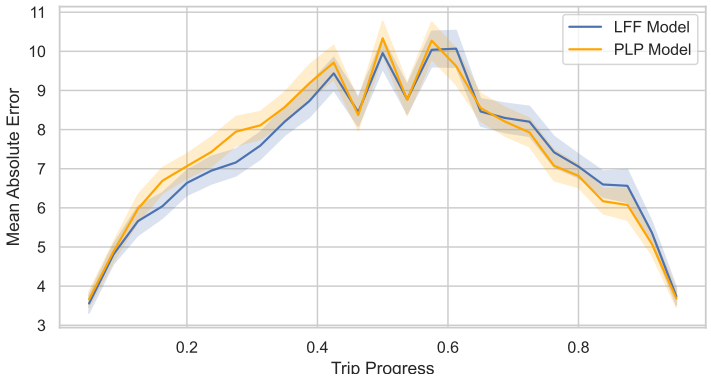

Figure 7.8: The MAE over varying values of the trip progress of the LFF model and the PLP model in the test set. The two plots show a 95% Confidence Interval computed using bootstrapping. The values of the trip progress have been binned in order to smooth the curve.

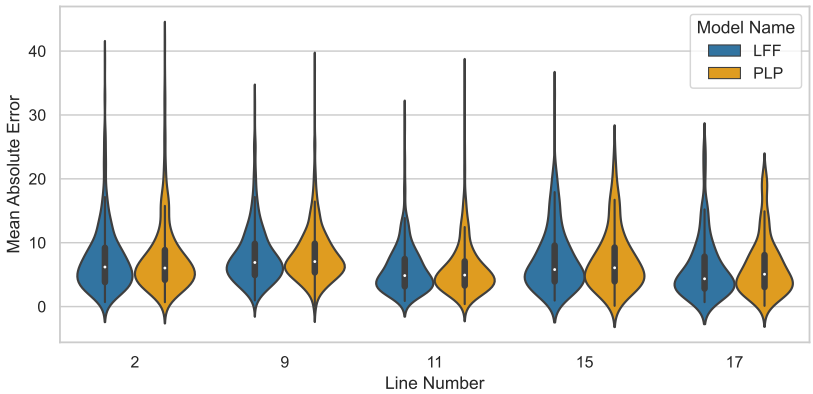

Figure 7.9: Violin plots of the MAE of the LFF and PLP models for each line number in the test set.

Metric	LFF Overall	LFF Load	LFF Flow
RMSE	11.60	12.78	17.33
MAE	7.83	8.58	11.22
$\mathbf{R^2}$	0.844	0.811	0.653

Table 7.4: 5-fold cross-validation metric scores for the LFF model including results of both of its load and flow components. The values shown are the mean scores.

Metric	Best Possible Score	Actual Score	Worst Possible Score
RMSE	8.10	11.16	17.13
MAE	5.17	7.52	12.68
$\mathbf{R^2}$	0.882	0.776	0.473

Table 7.5: Metric scores in the case that the fusion component would output the optimal prediction of the components, the actual fusion model's score and in the case that the fusion component would output the least optimal prediction of the components.

load component's distribution has a longer tail. This means that when it is more accurate than the flow component, the error difference is larger. This also explains the difference in metric scores in Table 7.4.

One of the hypothesised weaknesses of the flow component was that errors would accumulate throughout the trip due to the model's time-series manner of prediction. Figure 7.11 shows the trend of the mean error throughout a trip for both components. The flow component ostensibly has a higher error than the load component towards the end of the trip. This confirms the hypothesis that the errors accumulate. On the other hand, it seems that the flow component has a slight overall lower error in the first half of a trip. Fortunately, compared to the trend in Figure 7.8, it seems clear that the predictions by the fusion component do not suffer from the same trend as the flow component.

We have observed that the fusion component's metric scores are better than the scores of the individual components in Table 7.4. To identify the degree to which it improves upon both components, the following analysis is performed. For all instances in the test set, the best and worst of the two predictions by both components are selected and the metric scores are computed on these sets of the predictions. These scores are then compared to the metric scores of the LFF model. The fusion component is not a classifier that selects one of the two predictions, thus its performance could exceed these bounds. However, this analysis gives an idea of how "optimally" the current LFF model is able to aggregate the two predictions. The values of these scores can be found in Table 7.5. Note that this analysis is conducted on a single fold of the cross-validation experiment, hence the scores for the LFF model are slightly different compared to previous results. The table shows that the overall LFF score is quite close to the best possible score, compared to the worst possible score. However, it also indicates that the LFF model could still improve. In order to achieve the best possible score, given the current predictions of the components, the fusion model would need to improve its RMSE score by 27%, its MAE score by 31% and its R^2 score by 13%. This would require a significant better goodness-of-fit of the model and a stronger ability to generalise.

7.1.5. Discussion

The evaluation of the real-time models has shown that both the proposed LFF and PLP models outperform the random forest baseline significantly. Both models improve the metric scores of the baseline by at least 37%. The cross-validated results show that both models have very similar performance, but that the LFF outperforms the PLP model slightly and with a smaller standard deviation in the metric scores. The RMSE of the LFF model is 11.60 compared to 11.63 for the PLP model, which is only 0.03 lower. The MAE score is 7.83 compared to 7.87, another small difference. In terms of the R^2 metric, the models have the same score of 0.844.

Besides evaluating the regression scores, the ability to make crowding predictions has been evaluated as well using the same model predictions. Again, the LFF model is slightly better, both in terms of Macro F1 and Weighted F1, than the PLP model. The macro F1 score is 0.49, compared to 0.464,

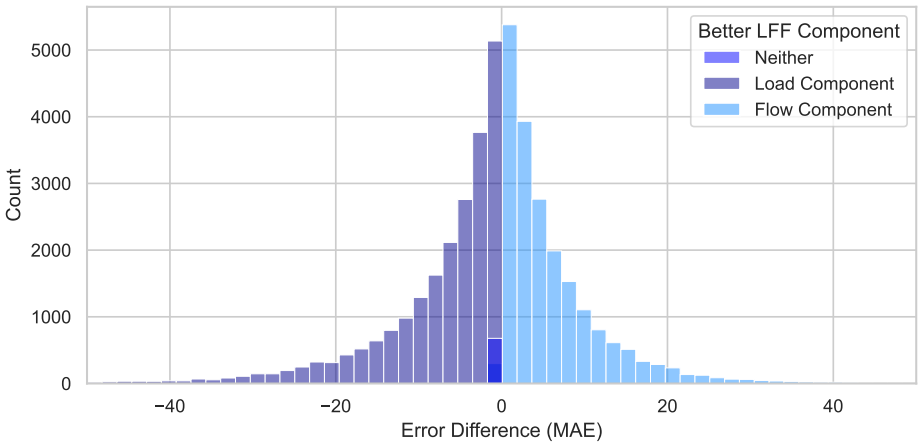

Figure 7.10: Distribution of differences in error between the load and flow components in the LFF model based on predictions on the test set.

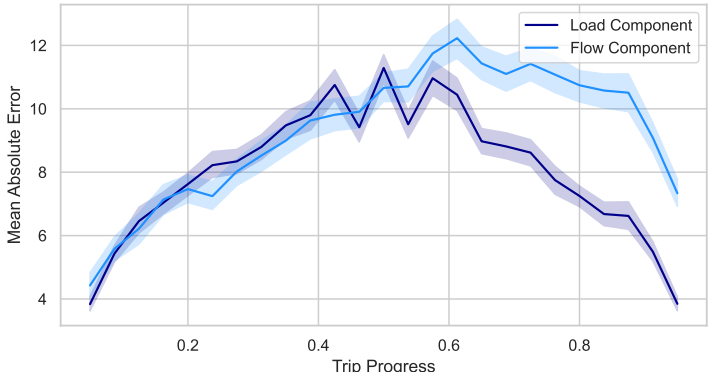

Figure 7.11: The MAE over varying values of the trip progress for the LFF model's load and flow components. The two plots show a 95% Confidence Interval computed using bootstrapping. The values of the trip progress have been binned in order to smooth the curve.

and the weighted F1 score is 0.828 compared to 0.823. Again, note that the differences are small. However, the difference compared to the baseline is similarly significant. In addition, both models outperform a dummy classifier, which predicts the most commonly occurring bin. Note that due to the class imbalance, the dummy classifier already scores a Weighted F1 score of 0.587. The performance by the LFF and PLP models is the highest for the lowest and most occurring bin, an F1 of at least 0.9, and it deteriorates significantly for increasingly higher bins. Note that the support for each bin also decreases and that these observations are related. It appears that the models seem to underestimate the passenger load for these larger bins. For incorrect classifications, the predictions are most often off by only one or two bins. This indicates that even incorrect classifications could be useful in a practical setting.

In the error analysis, it was shown that both the LFF and PLP models are competitive with each other and that the LFF has a better performance overall. The performance of both models significantly decreases for rare instances with large values for the passenger load and passenger flow. For these instances, the LFF model generally has a lower error than the PLP model. Note that these high-load and high-flow instances are extremely rare in the current dataset, due to the imbalance in the data. Therefore, the scores on these instances do not affect the overall metric scores significantly. Interestingly, it seems that for predictions over a trip that the LFF model initially outperforms the PLP model but that the LFF model's error is larger further down the trip. This is most likely due to the compounding errors in the flow component of the LFF model. Also observe that the error of both models is highest roughly midway through the trip, which is where peak passenger loads occur.

The analysis of the LFF model demonstrated that the fusion component outperforms the individual components. While the load component has a better overall error score than the flow component, the components outperform each other in roughly 50% of the cases. However, when the load component outperforms the flow component, it is to a more significant degree. The fusion model seems to exploit this phenomenon and create overall more accurate estimates. However, the fusion model does not optimally aggregate the two components' predictions and could theoretically still improve.

7.2. Real-Time Model Interpretability

In this section, we interpret the proposed LFF model by means of a SHAP values analysis as described in Section 3.6.2. First, the aggregated SHAP values are used to get a global view of feature importances in the model. Then, aggregated local explanations are used to understand how the values of individual features impact the output of the model. These values are computed for each component in the LFF model. The LFF model consists of three components: the load, flow and fusion components. The flow component consists of two sub-models: a board and an alight model. Hence, the LFF model has four sub-models for which the explanations need to be created. When referring to the models of the flow component, the terms flow-board and flow-alight are used to refer to either of the models. The larger the mean absolute SHAP value, the larger the impact of the feature on the model's output is. As all models under consideration are gradient boosting machines, the specialised TreeSHAP method by Lundberg et al. [49] is used to compute the SHAP values.

7.2.1. Global Feature Importances

Figure 7.12 shows bar charts of the mean absolute SHAP values for each feature in each of the sub-models of the LFF model. In this analysis, we refer to a feature being "important" when it has a relatively high mean absolute SHAP value. It is immediately obvious that the historical load/flow features have the largest impact on the outputs of the model. The monthly and weekly historical features are the top-two features for the models of the load and flow components. These historical features are more important by several factors compared to the other features. In addition, note that the combined importance of the historical features exceeds the importance of the vehicle-related features in the load and flow models. Surprisingly, the daily load feature is less important than the weight estimate in the load model. The historical features had a stronger relationship to the passenger load than any of the vehicle-related features in the feature analysis. Some other interesting features that are of influence are the time step of stop, day of the month and the door cycle count.

Interestingly, the board and alight models have a different ordering of important features despite having the same set of input features. It seems that features are related in a different sense to the alight count than to the board count. It also seems that the historical alight count features do not relate

to the historical board count features and vice versa. Moreover, the HVAC and weight estimate features do not have as strong a relationship to the output as the feature analysis indicated.

The fusion component's model appears to rely most heavily on the load component's prediction. The flow component's prediction seems to be significantly less important. This could be due to the overall better score of the load component, as can be seen in Table 7.4. The other features play only a marginal role in the prediction. Yet, the combined importances amount to a significant effect on the model's predictions.

7.2.2. Aggregated Local Explanations

The aggregated local explanations are shown through summary plots. These plots show the SHAP value for varying feature values for the top-20 features in terms of mean absolute SHAP values. It thus shows whether and to what degree the feature value has a positive or negative impact on the model output. In these figures, the thickness of the line indicates the density of SHAP values. Again, these figures are created for each sub-model of the LFF model. Figures 7.13, 7.14, 7.15 and 7.16 respectively display the summary plots for the load, flow-board, flow-alight and fusion model.

In the summary plot of the load model in Figure 7.13, it is clear that the monthly and weekly historical loads have a positive impact when their values are high, and a negative impact when their values are low. This makes sense as the features are strongly linearly related to the passenger load. However, observe how the daily historical load has an inverted effect. Somehow it seems to have the opposite effect on the output. It could indicate some interaction between those features, where the daily historical feature changes the output depending on the values of the other historical features. Note that the HVAC features have a very mixed effect, this could indicate some interaction with other features as well. It is also relevant to note that the door open time has a positive relationship to the passenger load, while it may seem unrelated at first glance. It also appears that the one-hot encoded line number feature has an effect on the output as well. A low value for line number 9 (i.e., the trip is not on line 9) appears to have a mixed effect on the output and a high value for line number 11 (i.e., the trip is on line 11) has a mostly positive impact on the output.

Similar to the summary plot of the load model, the summary plots of the flow-board and flow-alight models (figures 7.14 and 7.15) show that the historical board and alight count features have a mixed effect. Again, the weekly and monthly have a positive effect on the counts for higher feature values and the daily feature has the opposite effect. The flow-board model relies more heavily on the door open time whereas the board-alight model relies more on vehicle lateness. That is an interesting outcome, as the number of alighting passengers would intuitively not be expected to be affected by higher lateness values. The dwell times play a significant role as well, although it is difficult to say in what manner. The weight estimate plays a larger role in the flow-board model than in the flow-alight model, although the type of effect is similar. The door cycle count only seems to have a significant effect in the flow-alight model, despite it being to strongest feature in terms of correlation of the feature analysis. The headway, the third-highest feature in terms of correlation in the feature analysis, does not seem to play a significant role in either of the models. The remainder of the features seems to play a similarly minor role in the output of each of the models.

The fusion model does not incorporate any historical features and only uses the outputs of the two components and some additional features. From the summary plot in Figure 7.16 it is again clear how much the model relies on the load component's prediction. Both predictions have a positive effect on high input values. Also observe that the temporal factors play a significant role in the fusion model as well, as well as the line number features. It implies that the context plays a small, but noticeable role in how the model weighs the predictions of the load and flow components.

7.2.3. Discussion

The SHAP value analysis has made clear that the models in the components of the LFF model rely heavily on historical features. Vehicle-related features do play a significant role but it is a minor one compared to the various historical features. The relationships that historical features play may be dependent on one another, as some have an inverse relationship.

There are many differences between the importances of the features within the model and the results of the feature analysis from Chapter 5. Features such as the HVAC temperature delta and HVAC modes play a much smaller role in the model than the feature analysis may have implied. On the other hand, temporal features such as the time of day or the day of the month play a more significant

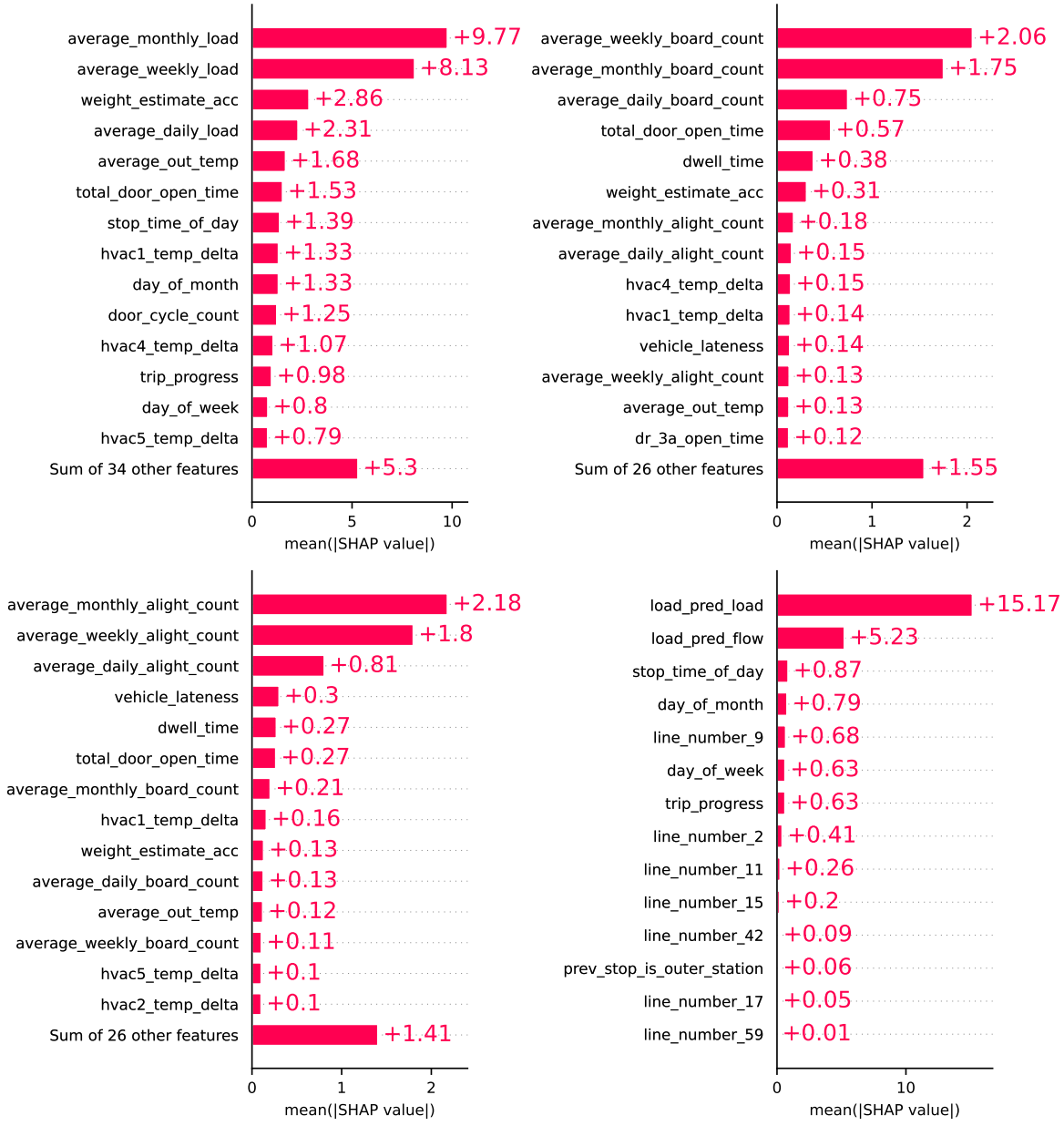

Figure 7.12: Barcharts showing mean absolute SHAP values across all test set instances for the different LFF sub-models: load component (top left), flow-board model (top right), flow-alight model (bottom left) and fusion component (bottom right).

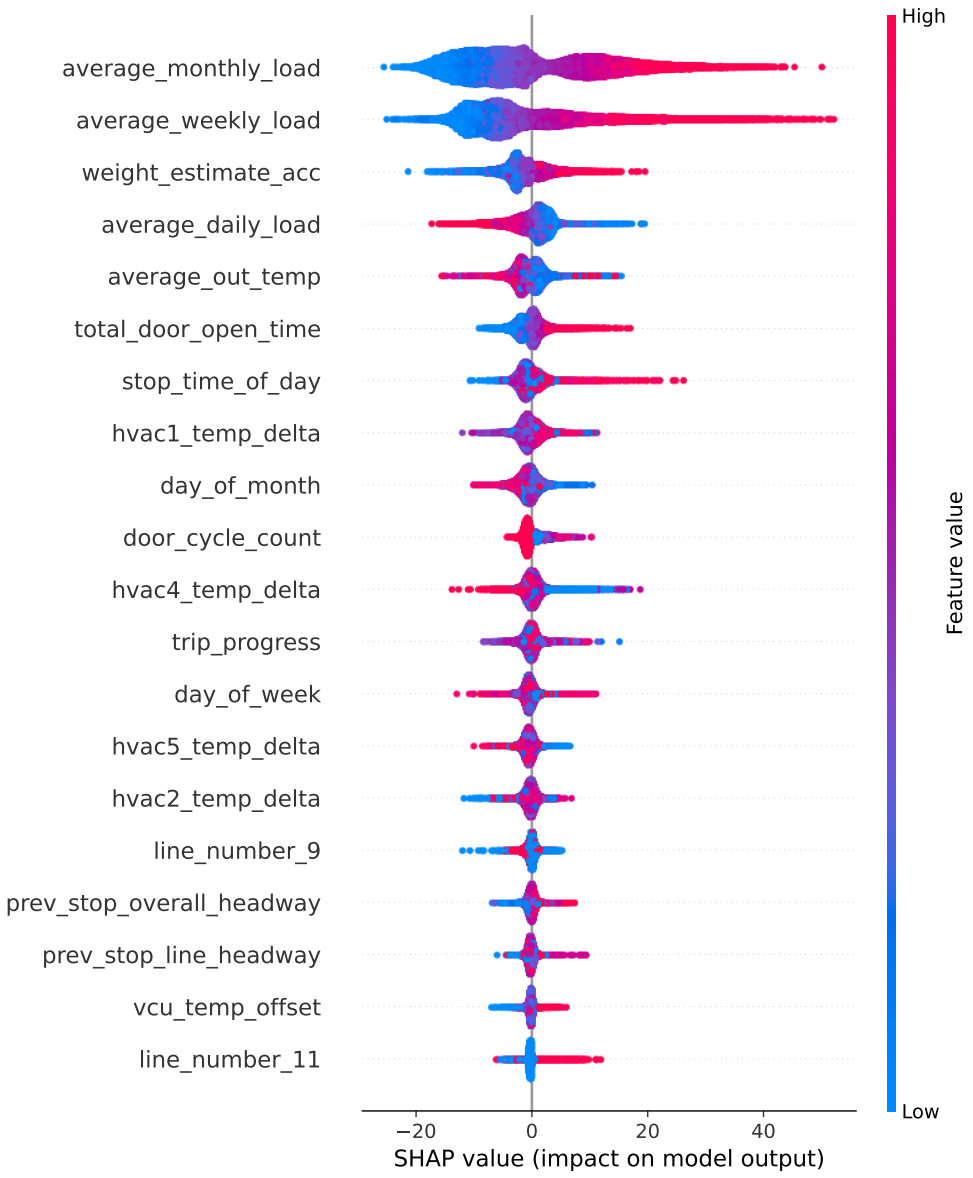

Figure 7.13: Summary plot of the test set SHAP values for the load model of the LFF Load Component. Red dots indicate a high feature value as input and blue dots indicate a low feature value as input. The thickness of the line in the figure indicates the density of SHAP values at that position.

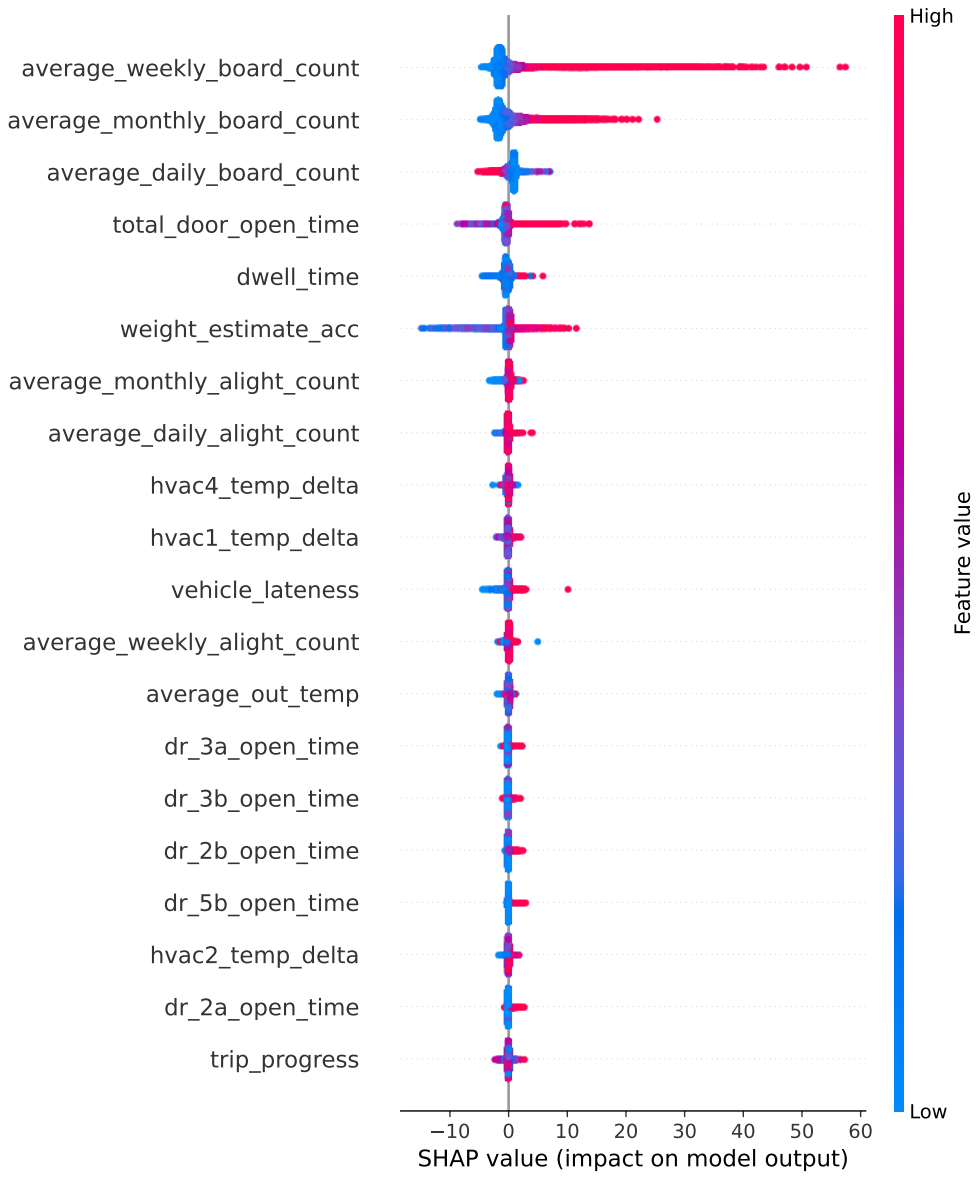

Figure 7.14: Summary plot of the test set SHAP values for the flow-board model of the LFF Flow Component. Red dots indicate a high feature value as input and blue dots indicate a low feature value as input. The thickness of the line in the figure indicates the density of SHAP values at that position.

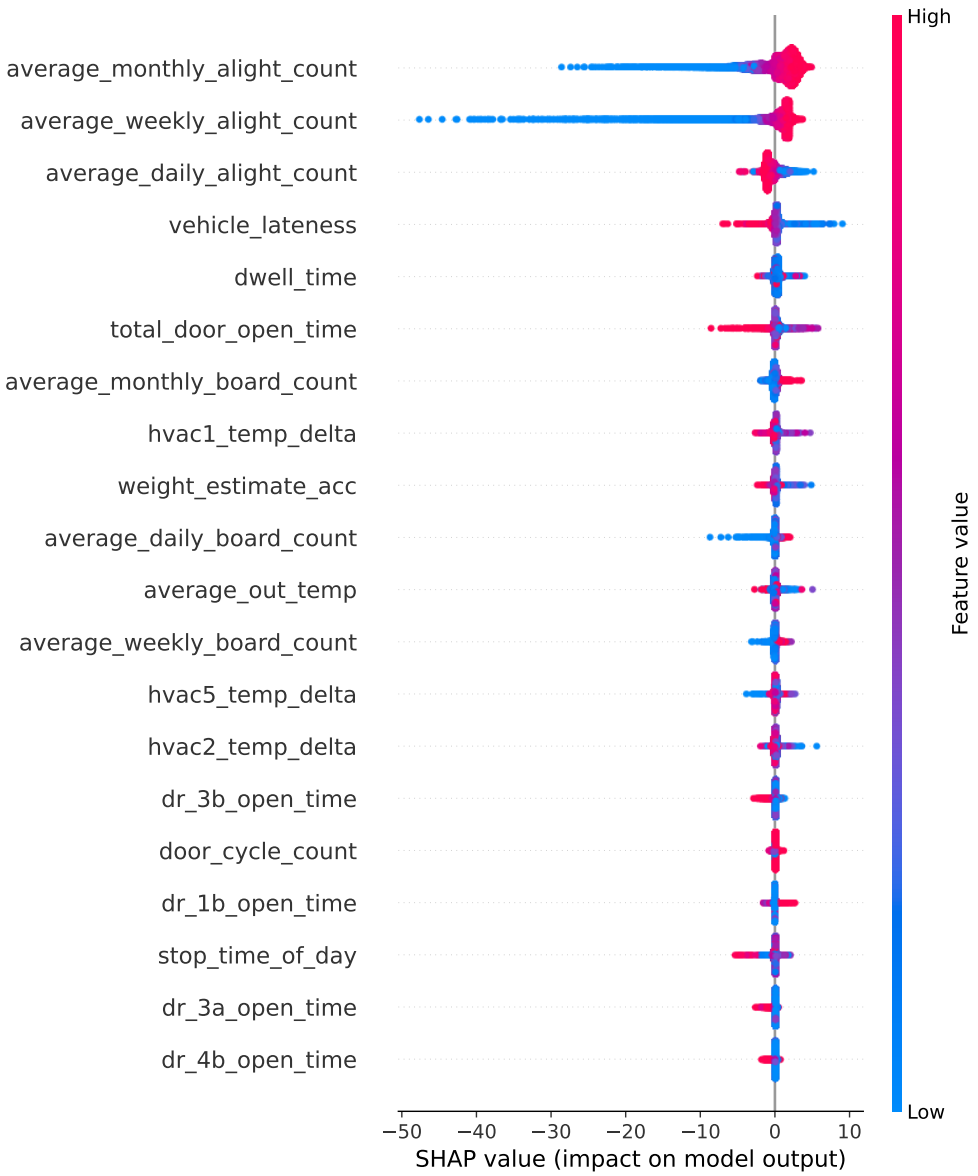

Figure 7.15: Summary plot of the test set SHAP values for the flow-alight model of the fusion model of LFF Flow Component. Note that alight counts are negative numbers, therefore a negative SHAP value means that the feature contributes to a *higher* alight count. Red dots indicate a high feature value as input and blue dots indicate a low feature value as input. The thickness of the line in the figure indicates the density of SHAP values at that position.

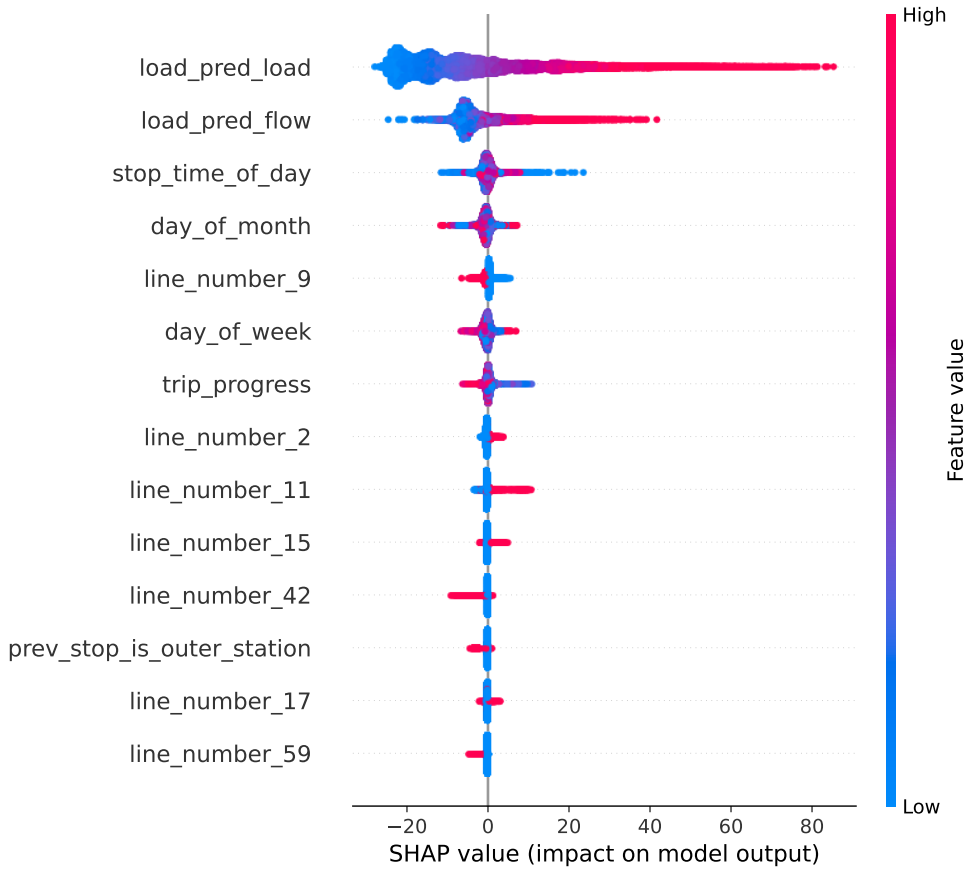

Figure 7.16: Summary plot of the test set SHAP values for the LFF Fusion Component. Red dots indicate a high feature value as input and blue dots indicate a low feature value as input. The thickness of the line in the figure indicates the density of SHAP values at that position.

role in the model compared to the results of the feature analysis.

The weight estimate, which was a very strong feature according to the feature analysis, also plays a significant role in the load and flow models. Other similarities to the results of the feature analysis are the relevance of the outside temperature in the load model, as well as the door-open times, trip progress and HVAC temperature deltas. For the flow models, the similarities to the feature analysis are the relevance of the door open times, the weight estimate and the HVAC temperature deltas.

Interestingly, the two flow models each rely on different features, while the output variables are closely related. The fusion model relies mostly on the load component's output, but the flow component's output still plays a significant role. Moreover, the temporal features have an effect as well, indicating that the fusion component takes the context of the prediction into account in weighing the two predictions.

Overall, the SHAP analysis has demonstrated that the expected relationships apparent from the feature analysis have only been partially upheld internally in the models. This is most likely due to their reliance on the historical passenger load and flow features. While the historical features are the predominant features in terms of importance, the results also show that the other vehicle-related features contribute to the prediction and that their combined contribution affects the prediction significantly. Therefore, we conclude that the vehicle-related features may have various mixed effects and when complemented by historical features, can be sufficient input to a predictive model.

7.3. Ablation Study

With the ablation study, we aim to quantify the contribution that the set of vehicle features makes and compare that to the contribution of the historical feature values. As described in Section 3.5, the most accurate model from the evaluation in Section 7.1.3 is re-evaluated using different sets of features as input. The result may indicate whether having only vehicle-related features offers a feasible alternative to historical AFC data.

The SHAP analysis of the previous section has already shown that the implemented LFF model has a great reliance on historical AFC features. Therefore, it is particularly relevant to investigate how well the model could perform without these features.

7.3.1. Experimental Setup

The same experimental setup as Section 7.1.2 describes is used. To reiterate, the implemented models are evaluated on the dataset using 5-fold cross-validation. The metrics used for evaluation are the MAE, RMSE and R^2 scores as defined in equations (3.4), (3.3) and (3.5). The mean score over the five folds is shown for each metric along with the standard deviation.

7.3.2. Results

Table 7.6 shows the results of the previously described experiment. Note that the results for the "LFF Complete" model are identical to the results for the LFF model in tables 7.1 and 7.4, as the same cross-validation procedure as used in those experiments has been applied.

The table shows that the complete model is optimal, followed by the model using only vehicle-related features and the model using only historical features. Based on the results in the table, excluding the vehicle features from the model yields an increase of 42% RMSE and 36% MAE and a decrease of 18% R^2. Excluding the historical features from the model yields a model with a decrease in performance of 16% in terms of RMSE and MAE and 6% in terms of R^2. Finally, the vehicle model's RMSE score, compared to the historical model, is 18% lower, the MAE score is 14% lower and the R^2 score is 15% higher.

The results of the same ablation study on the PLP model can be found in Appendix F.

7.3.3. Discussion

The results in Table 7.6 show that having the complete set of features is optimal, as it yields the most accurate model. However, when having only a subset of the features, it seems that the vehicle-related features provide a better model compared to the historical features.

Excluding either set of the features has a negative effect on the model's performance. However, the decrease in performance is less significant for excluding the historical features, compared to excluding the vehicle-related features. In fact, the vehicle model performs up to 18% better, in terms of RMSE.

	LFF Complete		LFF Vehicle		LFF Historical	
	Mean	SD	Mean	SD	Mean	SD
RMSE	*11.60*	0.189	13.43	0.205	16.43	0.126
MAE	*7.83*	0.082	9.12	0.119	10.66	0.098
$\mathbf{R^2}$	*0.844*	0.012	0.792	0.008	0.688	0.018

Table 7.6: Experimental results on evaluating the different feature sets on the LFF model. The LFF Vehicle column contains the results for the LFF model using all vehicle-related features and the LFF Historical column contains the results for the LFF model using all historical features. All results show the mean metric values with standard deviations of 5-fold cross-validation. The best mean scores are written in italics.

The SHAP analysis of the previous section has shown that the LFF model relies mostly on historical features for its predictions. However, the current ablation study demonstrates that, in the absence of historical features, the vehicle-related features contribute to an adequate model that outperforms its counterpart: a model with only historical features. This result underlines the contribution that the vehicle-related features make to the model.

It is not appropriate to claim that these results mean that the vehicle-related features are preferable to the historical features in general, this claim only holds in the current context of the LFF model. However, it is safe to assume that the vehicle-related features offer a reasonable alternative to the historical features in a general sense.

7.4. Real-Time Dashboard

In order to emphasise the practical implications that the results of this research may provide, a dashboard has been developed as a proof-of-concept product that displays real-time estimations of the passenger load on the transport network. The estimations shown in the dashboard are constructed using real-time data and predictions by the LFF model. Such a dashboard could be used by transport operators to acquire insights into the current state of the transport network which may facilitate ad hoc decision making.

7.4.1. Implementation Details

While the data engineering pipeline as described in Chapter 4 was implemented to extract a post hoc evaluation dataset, similar techniques are used to extract the data for the real-time dashboard. In a real-time setting, only the vehicle and the GTFS data are available. This simplifies the matching procedure of the data sources. The GTFS data contains a real-time update stream that can be accessed and includes the latest updates on currently active trips. This data contains a reference to the vehicle number which can be used to retrieve the relevant vehicle data. Then, a similar procedure is described as in Section 4.4 can be used to match the data sources using the vehicle's GPS coordinates and the GPS coordinates of the station. Relevant data regarding the station is given by the GTFS stream, as its updates contain references to that information in the static GTFS data. Finally, the features are extracted using the same methods as described in Section 4.5.

Once the data has been extracted for a given trip, a model such as the LFF model can be used to predict the passenger load. The resulting data, including the set of predictions, is saved and is used as input to the dashboard. This dashboard can update based on incoming data and thus provide the user with real-time visual aids to get insight into the passenger load on the network. Earlier predictions can be shown as well, simply by loading archived predictions.

The data extraction pipeline was implemented in Python [72] and the dashboard was created using Tableau[3]. The execution of the whole pipeline, including the estimation of the passenger load, has an average duration of 90 seconds. Thus, the visualisation results on the dashboard have a slight delay. A screenshot of the dashboard is shown in Figure 7.17.

7.4.2. Evaluation

In a real-time setting, it is difficult to evaluate the predictions that are made by the model as there is no AFC data containing ground-truth labels in the current context. However, to demonstrate the

[3] https://www.tableau.com/

Figure 7.17: Screenshot of the real-time dashboard. It displays the passenger load for active trips over the network on a map, as well as a pie-chart of the peak occupancy rate over the vehicles and area charts with the passenger load over the vehicles for the past trips. The actual vehicle numbers have been replaced by arbitrary numbers.

effectiveness of the implemented real-time dashboard, we have retrieved a small AFC dataset over a period of four days and evaluated the real-time predictions of the LFF model. The set of real-time data over that period contained 288 trips with 6,719 rows in total. Filtering out faulty trips and missing data yielded an evaluation dataset of 231 trips with 5,002 rows in total. The LFF model was fitted to the whole dataset of the previous analyses, including the test trips. The training dataset thus consisted of 12,204 trips with 291,486 rows. Table 7.7 shows the results of the evaluation using the same metrics as the previous evaluations.

Observe that the metric scores have deteriorated compared to the original real-time model evaluation in Table 7.1. While the RMSE score is only increased by 0.87 and the MAE by 0.78, the R^2 score has decreased to 0.429. That is a reduction of 49%, which merits further investigation.

A low R^2 score indicates that there is a large amount of variance in the data that could not be explained by the model. In this case, it ostensibly indicates that there is a significant amount of unexplained variance in the passenger load. One probable reason for this is that the data used to train the model was collected over a period during which pandemic-related restrictions were in place due to the outbreak of COVID-19, affecting public transport usage [3]. An inspection of the distribution of the passenger load over the two datasets, compared to the distribution of the passenger load predictions by the model on the current dataset confirms a difference in distribution. The distributions are shown in Figure 7.18. The original dataset used for training has a thinner and long-tailed distribution while the current dataset has a shorter but more fat distribution. The figure clearly shows that the model is not fully able to capture this wider region and the distribution seems to fall between the two other distributions. A difference in distribution may be explained by the specific time period of the real-time data sample, which is early April. However, as Figure 7.19 shows, the difference is even more emphasised in the specific date range in both datasets.

This may indicate that the distributions are inherently different and that there may be a so-called pandemic-related effect in the data. However, it does not rule out all other possible factors that may lead to a difference in the distributions. To find a definitive conclusion to this hypothesis, a comprehensive analysis of the data is required. As this is out of scope for the current research, this shall be left for future work. Based on the results thus far, we conclude that the reduction in R^2 is suspected to be due to the difference in traveller patterns in accordance with the alleviation of the pandemic-related restrictions.

Metric	LFF Score
RMSE	12.47
MAE	8.61
$\mathbf{R^2}$	0.429

Table 7.7: Metric scores of the evaluation of the LFF model on a sample of data collected by applying the model in a real-time setting between 31 March 2022 and 4 April 2022.

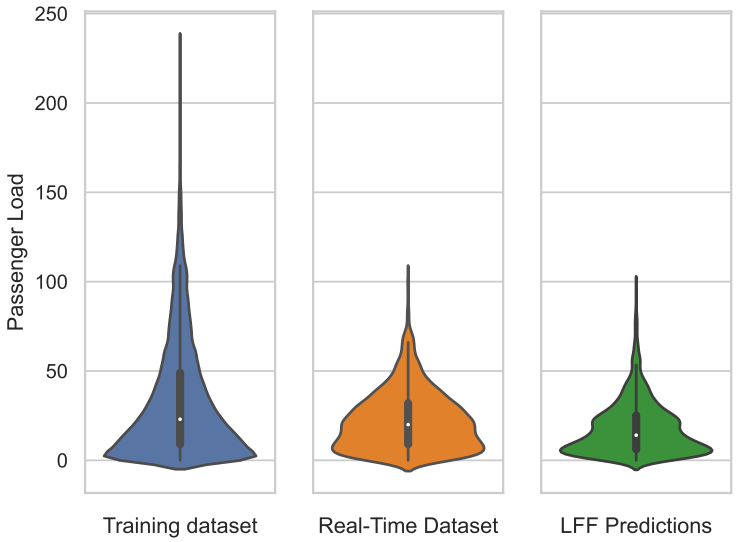

Figure 7.18: Distribution of the passenger load of instances in the evaluation dataset of the real-time proof-of-concept product showing the distribution of the training dataset (left), the evaluation dataset (middle) and the LFF model's prediction on the evaluation dataset (right).

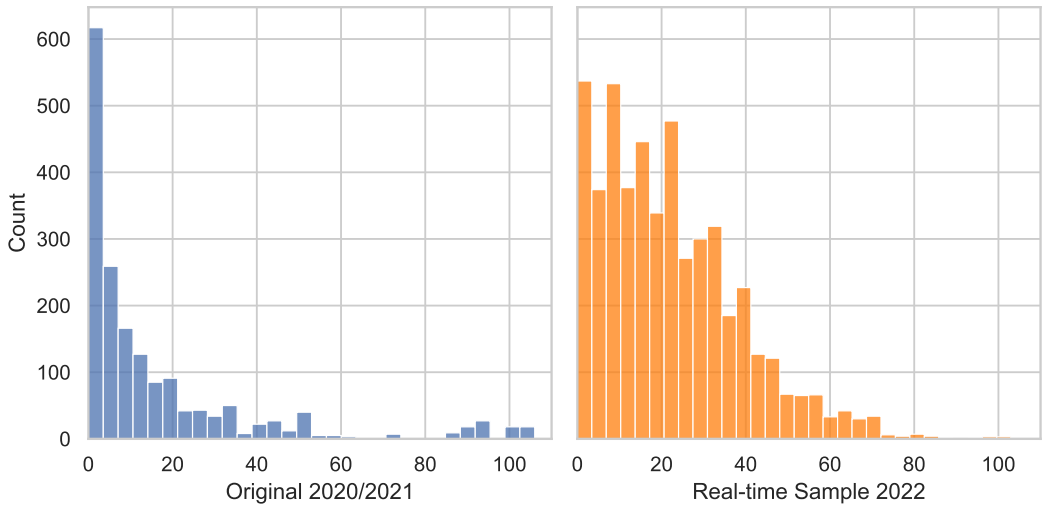

Figure 7.19: Distribution of passenger load values in both the original and real-time sampled dataset filtered to contain only data between 31 March 2022 and 4 April 2022.

7.4.3. Discussion

Overall, the proposed real-time dashboard provides a proof-of-concept product of how the results of the current research can be applied in practice. Moreover, the dashboard may even be extended to provide short-term forecasts using the Seasonal ARIMA with GARCH and exogenous variables model presented in Section 6.2. This may allow the dashboard to provide additional short-term forecasts to the user. Further use cases of the dashboard are proposed in Section 8.4.

The evaluation demonstrated that the RMSE and MAE scores of the LFF model predictions were not greatly affected, while the R^2 score was reduced by 49%. A brief analysis indicated that this is due to a difference in the distribution of the passenger load in the evaluation dataset. This difference in distribution is suspected to be due to the presence and absence of pandemic-related restrictions during the time periods of training and test datasets. Further research should aim to analyse these differences in distributions and provide a definitive answer to the presence of a pandemic-related effect in the dataset.

7.5. Forecasting Model

In this section, we perform the evaluation of the forecasting model as proposed in Section 6.2 according to the methodology described in Section 3.6.3. First, we provide the experimental setup, followed by the results of the evaluation. Finally, we provide a discussion of the results. While the individual performance of the models is primarily evaluated, we also consider how incoming real-time signals contribute to improving the accuracy of the forecasts. We also evaluate how model-predicted labels contribute to a better forecast, which is relevant in a practical setting. To avoid confusion for the reader, the results of this latter evaluation are provided in a separate subsection from the main results.

7.5.1. Experimental Setup

As time-series models require different parameters depending on the nature of the time series, a dataset is selected based on a single tram route to be used as the evaluation scenario. In the current evaluation, the tram route of line 11 from “Rijswijkseplein” to “Scheveningen Haven” is used. Figure 7.20 shows the mean passenger load trend on this route. This subset of the data is split into a training portion and a test portion. The random forest baseline is fitted to the training set and evaluated on the test set. The forecasting model will fit a subset of the training set, taken as a “history” to the test set trip under evaluation. Some knowledge of the trip under consideration is used to select this history, such as the start time of the trip. However, the ground-truth labels or any of the real-time features are not used to select this history.

The current models collect a history of trips that started at the same time step of the day as the current trip under evaluation. If an insufficient amount of trips are found to yield a minimum amount of 400 observations as history, additional trips are considered that started a time step earlier or later. If this does not yield enough observations, then the search iteratively broadens to trips on earlier and later time steps until enough observations have been collected. If more observations have been collected than the upper bound of 600 observations, trips are randomly discarded until the number is below the upper bound. These bounds on the number of observations have been set based on exploratory analysis to avoid overfitting the model to the training data and to promote sensitivity to the incoming observations.

The forecasting model will iteratively forecast the passenger load over the entire trip. After an iteration of forecasts, the model is updated by the observed passenger load. This can either by a ground-truth label or, in a practical setting, a real-time model prediction. In the current experiment, both approaches are performed. The ground-truth labels are provided to purely investigate the performance of the forecasting models and their ability to update and the model-predicted labels are used to simulate a more realistic setting. Note that the model-predicted labels, as provided by the LFF model, contain some error which affects the performance of the forecasting models. The history of trips to which the forecasting models are fitted contains only ground-truth labels.

Once one iteration of forecasts has been performed and the model has been updated with the current stop’s label, then the passenger load of the remaining stops is forecasted again. This process continues until the final stop and it is repeated for each of the trips in the test set. An illustration of the current experimental setup is shown in Figure 7.21.

As described in Section 6.2, the base model for forecasting is the Seasonal ARIMA model. Several

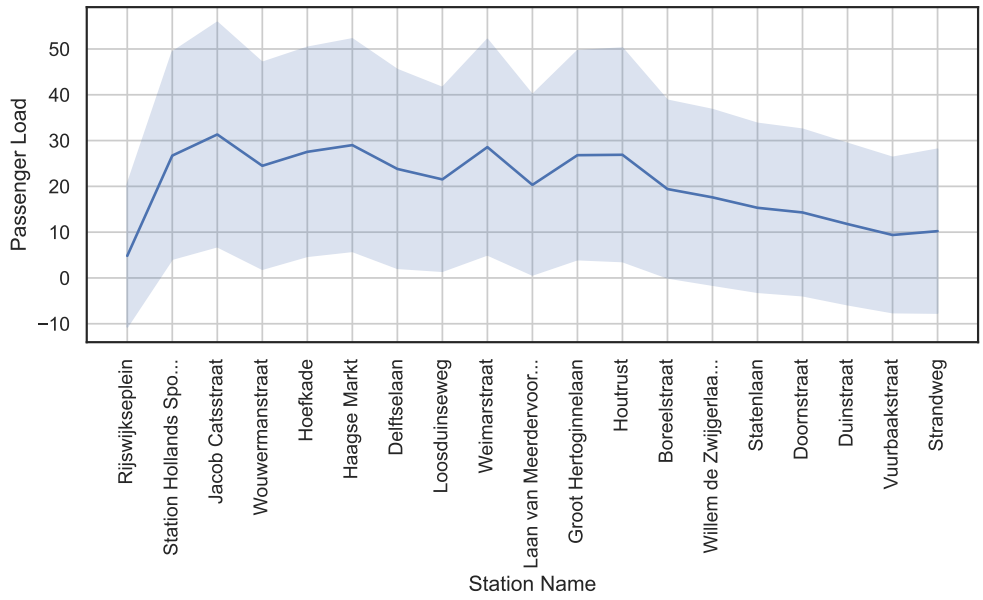

Figure 7.20: Trend of passenger load during trips in the forecasting dataset on line 11 from ”Rijswijkseplein” to ”Scheveningen Haven”. The mean passenger load is shown with a standard deviation for all trips in the dataset. Station names are shown in the order of the trip.

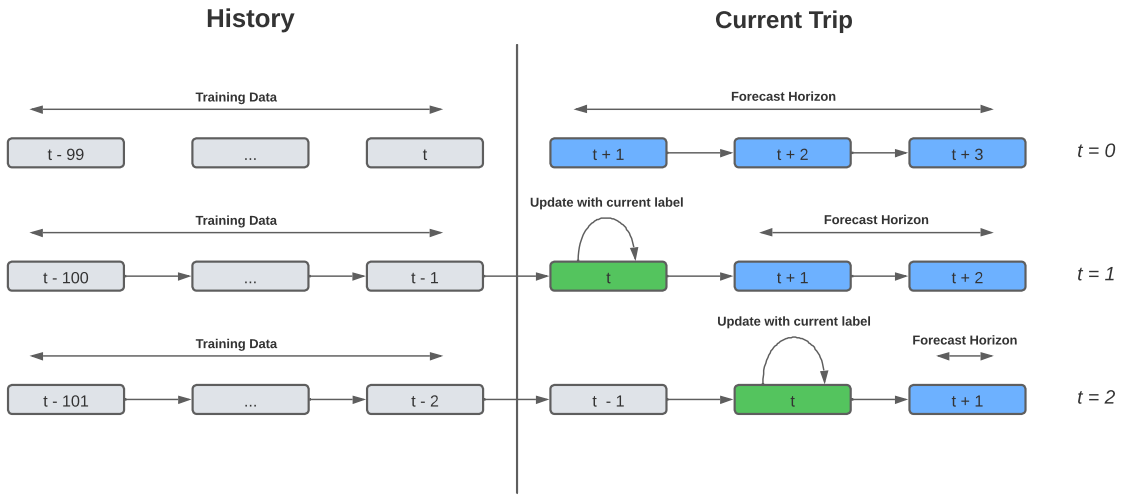

Figure 7.21: Illustration of the forecasting experiment of a single trip where each row in the figure shows a single iteration of the experiment. The grey boxes indicate past time steps, the green boxes indicate the current time step and the blue boxes indicate future time steps. The model forecasts the remaining from the current time step after updating itself. The first row lacks a current time step as it is the initial forecast of the trip.

Forecasting Model	MAE	IMP
SARIMA	6.68	-1.026
SARIMA + GARCH	6.30	-0.992
SARIMA + EXOG	5.24	-0.943
SARIMA + GARCH + EXOG	5.16	-0.876
Random Forest Baseline	7.16	–

Table 7.8: Overall Mean Absolute Error (MAE) and Mean Improvement (IMP) scores for all forecasts for each of the forecasting model variations. The baseline does not contain an IMP score as it is not updated over the forecasting horizon. The names of the forecasting models indicate the specific configuration of the forecasting model, where the plus sign indicates a combination of models. Seasonal ARIMA is abbreviated to SARIMA. Generalized AutoRegressive Conditional Heteroskedasticity is abbreviated to GARCH. Finally, linear regression using exogenous variables is abbreviated to EXOG.

variations, incorporating GARCH and exogenous variables, are evaluated. Both variations are integrated into the Seasonal ARIMA model and evaluated as well as the combination of them added to the Seasonal ARIMA model. The parameters for the Seasonal ARIMA model are selected using an automated search through the pmdarima library [69]. It performs a grid search on various parameters and optimises towards the best Akaike Information Criterion score.

The metrics for evaluation include the Mean Absolute Error (MAE) as defined in Equation (3.4), as well as the Look-Ahead Error (LAE) from Equation (3.7) and the Mean Improvement (IMP) as defined in Equation (3.8). As discussed previously, updates using both the ground-truth labels as well as model-predicted labels are considered. The model-predicted labels are provided by the LFF model. The model is trained on the training data of one arbitrary fold from the real-time cross-validation evaluation in Section 7.1, excluding any test trips from the current test set.

The current evaluation is mainly focused on updates using the ground-truth labels, to ensure that the analysis is restricted to the performance of the forecasting models rather than the LFF model. As mentioned previously, the results for updates using the LFF-predicted labels are provided in a separate section.

The set of “history” data contains 1,193 trips containing a total of 22,667 rows and the test set contains data of 114 trips, resulting in 2,166 individual rows.

7.5.2. Evaluation Results

We present the results of executing the previously described experiment in this section. Table 7.8 shows the aggregated scores for the MAE and IMP metrics for all forecasts made in the experiment, regardless of the forecasting horizon.

Figure 7.22 shows the MAE for each station, both for initial predictions and the latest time step predictions. Note that the errors for predictions after the initial predictions but before the latest predictions are most likely located within the region bounded by the lines in both figures for each model, considering that the model fine-tunes itself with incoming updates after each prediction. Figure 7.23 displays the look-ahead error over different forecasting horizons. Figure 7.24 displays the mean improvement for the varying models across different stations.

When considering overall forecasting performance, it is clear that the combined model of Seasonal ARIMA with GARCH and exogenous variables is the best model. It has the lowest overall MAE and LAE scores. The second-best model is the Seasonal ARIMA with exogenous variables. Notice that adding GARCH to the model only makes a very minor difference in the MAE score, both for the case of Seasonal ARIMA on its own and Seasonal ARIMA with exogenous variables.

It is interesting to note that for all stations, the error drops far below the baseline’s error for the latest predictions. It is clear that the models are able to update very well to the incoming signals. However, this does not seem to hold for station “Station Hollands Spoor” which coincidentally has the highest mean passenger load of all stations, as can be seen in Figure 7.20. It seems that after that station, the models’ forecasts improve significantly. This is evident both in figures 7.22 and 7.24. It appears that the station “Station Hollands Spoor” is most difficult to predict but that it provides the largest improvement load for the rest of the trip. Thus it seems that signals from that station provide the most information that the models can learn. The improvement for later prediction visible in Figure 7.22 is more clearly apparent in those figures.

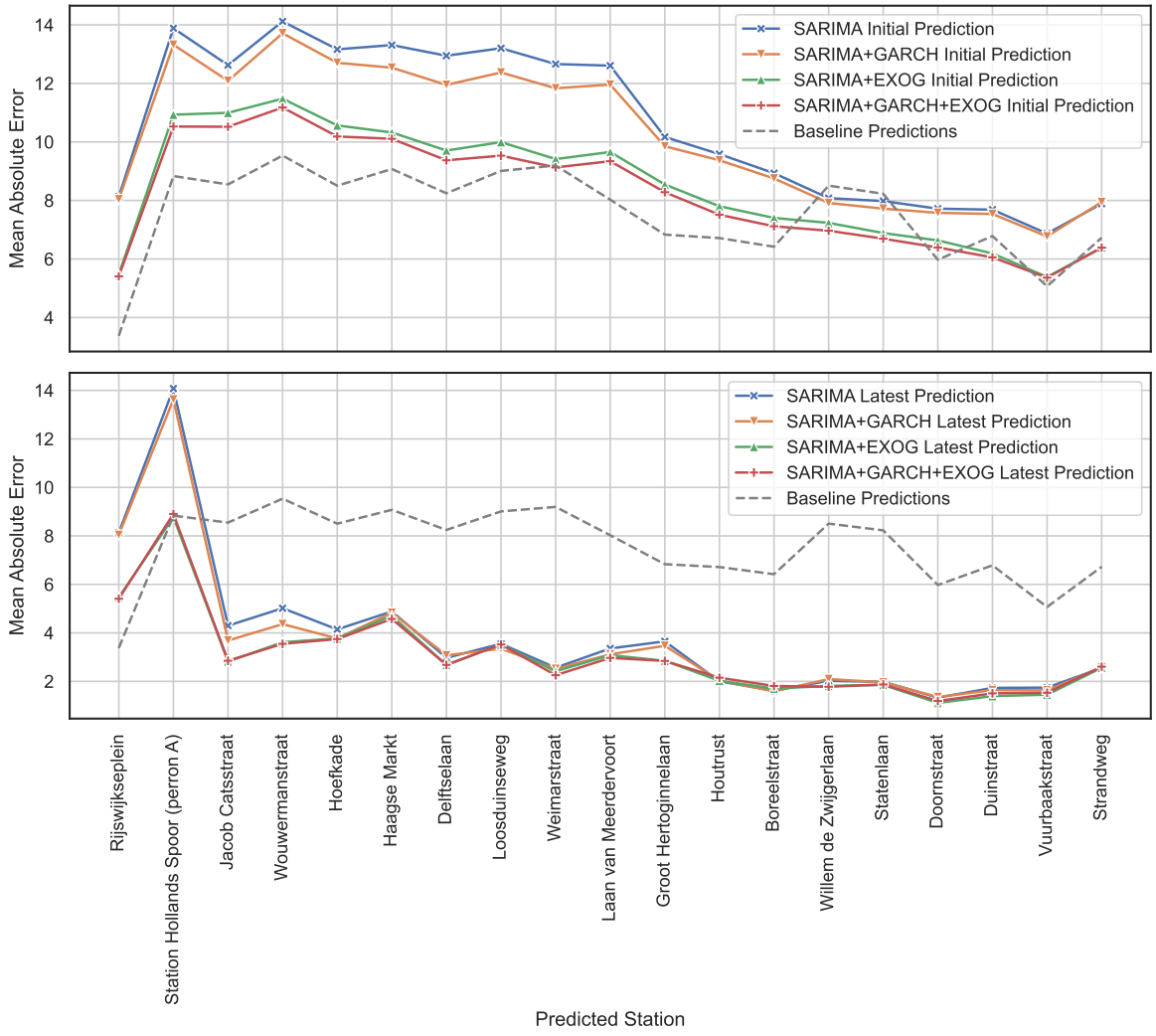

Figure 7.22: Mean absolute error for each station by each forecasting model variation showing initial predictions (top) and look-ahead-1 predictions (bottom). The baseline model's predictions are included in both figures. Stations are in order of the trip. Seasonal ARIMA is abbreviated to SARIMA. Generalized AutoRegressive Conditional Heteroskedasticity is abbreviated to GARCH. Finally, linear regression using exogenous variables is abbreviated to EXOG.

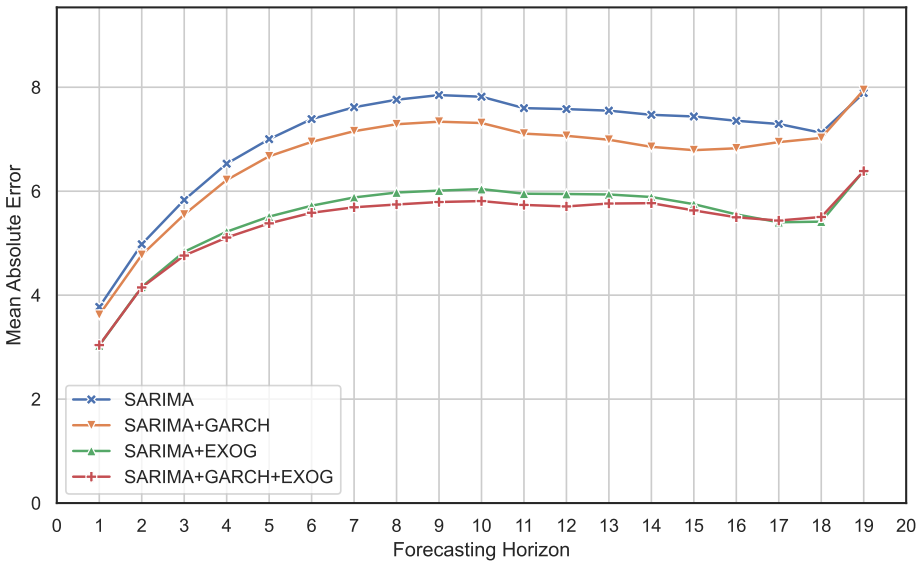

Figure 7.23: Look ahead error over varying horizons for the different forecasting model variations. Seasonal ARIMA is abbreviated to SARIMA. Generalized AutoRegressive Conditional Heteroskedasticity is abbreviated to GARCH. Finally, linear regression using exogenous variables is abbreviated to EXOG.

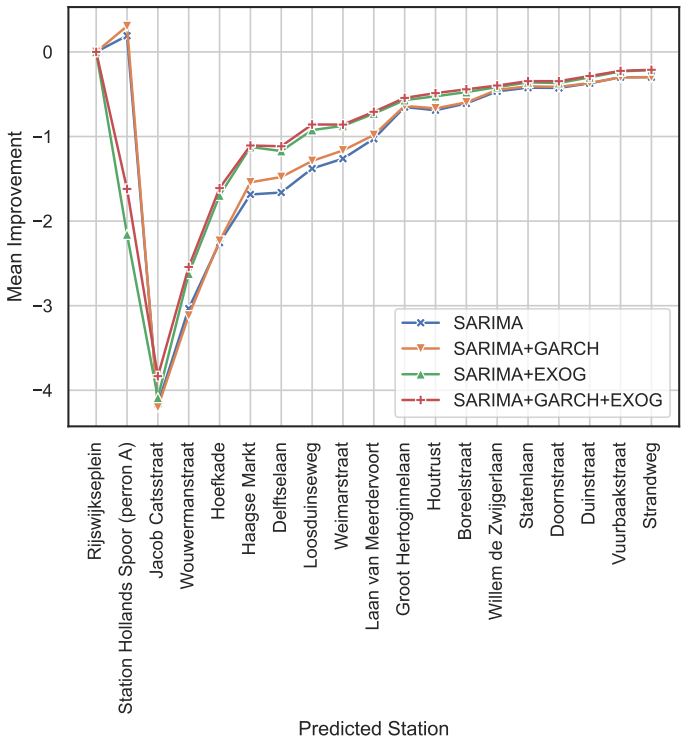

Figure 7.24: Mean improvement per station for the different forecasting model variations. Stations are in order of the trip. Seasonal ARIMA is abbreviated to SARIMA. Generalized AutoRegressive Conditional Heteroskedasticity is abbreviated to GARCH. Finally, linear regression using exogenous variables is abbreviated to EXOG.

The Seasonal ARIMA model has the largest overall improvement. This is most likely due to its relatively low goodness-of-fit and that it can therefore improve its performance more than the other models. For all models, it seems to be that improvements suffer from diminishing returns such that at the end of the trip, the models improve only slightly. The overall improvement is mostly below zero, indicating a decrease in the absolute error. For the SARIMA and SARIMA and GARCH models, the mean improvement score is larger than zero for “Station Hollands Spoor” which indicates that the error decreases slightly.

The look-ahead error in Figure 7.23 indicates that a larger forecasting horizon indicates a larger error. The error seems to increase every time step but reaches a peak before decreasing slightly. This could be due to the lower variance that the later stations seem to have in Figure 7.20.

Several examples of how the models change their forecasts based on the update signals can be found in Appendix G. Detailed results of the MAE score per station and forecasting time step are provided in Appendix H. Finally, a brief discussion of the fitting and prediction times of the models is provided in Appendix I.

7.5.3. Model-Predicted Labels

In this section, we explore how using model-predicted labels as real-time observations influences the accuracy of the forecasts. As described previously, the model predictions by the LFF model are used as real-time signals rather than the ground-truth passenger load labels. Note that the LFF model’s prediction had an overall MAE score of 5.27 on the current test set, where “Station Hollands Spoor” had the highest MAE of 6.80 and station “Rijswijkseplein” had the lowest MAE of 2.68.

The results of the evaluation of using both the ground-truth label and the LFF-predicted labels as updates for the forecasting models are shown in Table 7.9. Note that the scores of the models using ground-truth labels are identical to the results shown in Table 7.8, the results have been duplicated in this table for the reader’s convenience.

Observe that while the results of the forecasting models are overall worse for updates using the LFF-predicted labels, the difference is not as substantial as one might expect. On average, the overall MAE is increased by only 1.18 for each model. The models incorporating exogenous variables outperform the baseline overall, with the model incorporating exogenous variables as well as GARCH having the best performance. Given the MAE of the LFF-predicted labels, it indicates that the models can still use these labels to derive a better forecast. While the IMP scores are worse, 0.405 higher on average, the results show that the models still improve when given a real-time passenger load estimate by the LFF model. The trend of this improvement is further investigated. As the results are expected to be similar for all models, the results of only one model are considered, namely the Seasonal ARIMA with GARCH and exogenous variables model. Figure 7.25 displays the mean improvement per station, aggregated for all instances in the test set. Again, the results for the model using ground-truth label updates are duplicated. In this case, the results from Figure 7.24 are reused, hence the red colour in the current figure. The figure clearly shows that the LFF-predicted labels cause an improvement for all stations, although less significantly than the ground-truth labels. The trend of diminishing returns is similar for both lines, indicating that improvements in error increasingly get smaller further down the vehicle’s trip.

However, the improvements are not monotonic. In other words, an improvement in the error by means of the incoming observations is not guaranteed. This is demonstrated by Figure 7.26, which shows the distribution of individual improvement values on forecasts. The distributions of the improvement by updating using either type of label have an expected value below zero, but positive values exist as well. For the LFF-predicted labels, in particular, the distribution is shifted towards positive values compared to the ground-truth label’s distribution. This indicates that while the incoming observations improve the performance overall, as the expected value is below zero, increases in error occur as well. This is especially true for the LFF-predicted labels.

7.5.4. Discussion

Overall, it is clear that the forecasting models based on Seasonal ARIMA are suitable for passenger load forecasting. While a Seasonal ARIMA incorporating both GARCH and exogenous variables is optimal, the GARCH component only adds a marginal improvement to the forecasting accuracy and the exogenous variables provide the best increase in accuracy. All model variations show improvements in accuracy based on incoming real-time signals and thus demonstrate that the real-time signals can be effectively used for real-time passenger load forecasting. However, the improvements caused by

Model Type	MAE		IMP	
	Ground Truth	LFF	Ground Truth	LFF
SARIMA	6.68	7.79	-1.026	-0.632
SARIMA + GARCH	6.30	7.40	-0.992	-0.599
SARIMA + EXOG	5.24	6.56	-0.943	-0.523
SARIMA + GARCH + EXOG	5.16	6.35	-0.876	-0.463
Random Forest Baseline	7.16	7.16	–	–

Table 7.9: Overall Mean Absolute Error (MAE) and Mean Improvement (IMP) scores for all forecasts for each forecasting model using either the ground-truth passenger load labels for updating or the predicted labels by the LFF model. The baseline does not contain an IMP score as it is not updated over the forecasting horizon. The names of the forecasting models indicate the specific configuration of the forecasting model. Seasonal ARIMA is abbreviated to SARIMA. Generalized AutoRegressive Conditional Heteroskedasticity is abbreviated to GARCH. Finally, linear regression using exogenous variables is abbreviated to EXOG.

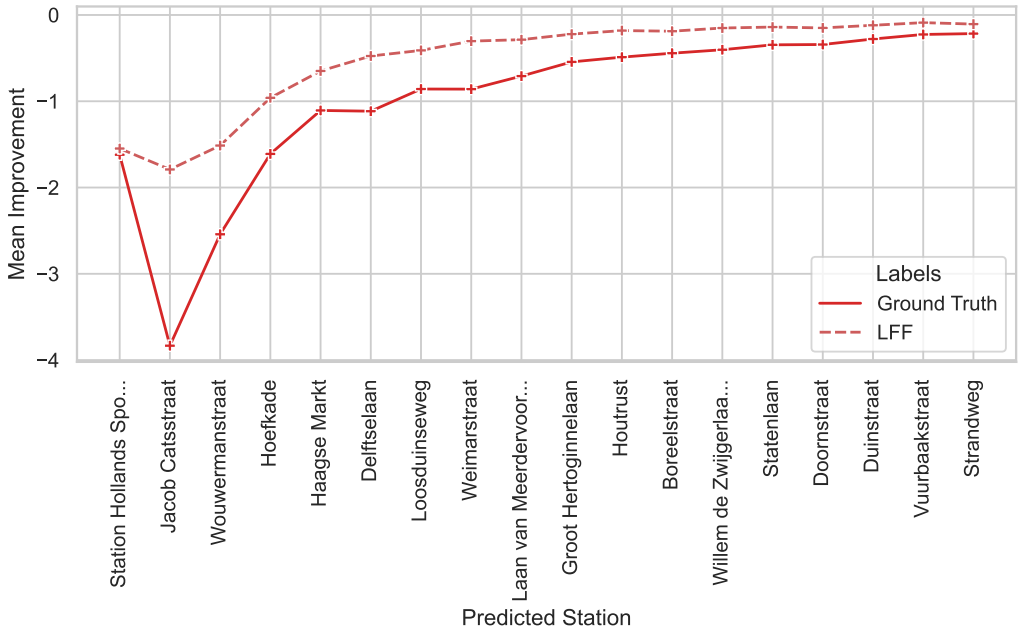

Figure 7.25: Mean improvement per station for the Seasonal ARIMA model with GARCH and exogenous variables using either ground-truth or LFF-predicted labels. The results are collected on a test set of 114 trips. The station names are shown in the order of the trip.

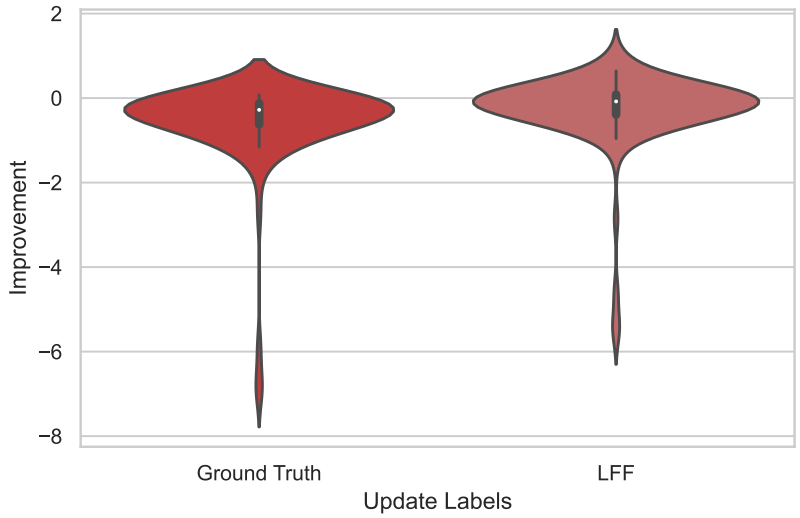

Figure 7.26: Violin plots of improvement (IMP) values for individual trip forecasts of the evaluation using two different update labels for the Seasonal ARIMA model with GARCH and exogenous variables.

these incoming signals suffer from diminishing returns.

It is clear that models perform optimally when forecasting a short horizon. This highlights the contribution that incoming signals can make. Effectively, by updating the models in real-time the forecasting horizon is reduced, also for stops further down the trip.

While we expected that updating the models using LFF-predicted labels would significantly decrease performance, it appeared to have a much less significant negative impact. The models are still able to improve their predictions and even outperform the baseline. This demonstrates the practical applicability of the models, particularly the Seasonal ARIMA model incorporating both GARCH and exogenous variables, in a setting where real-time ground-truth labels are not available. However, we also observe that the variance in improvements is large and that the error can also increase for incoming observations.

8

Conclusion

To conclude this research, we provide definitive answers to the research questions. In addition, we provide a discussion regarding the implications, limitations and proposed future work of this research that places the findings into a broad scientific and practical context.

8.1. Summary of Findings

The goal of this research work has been to investigate how real-time vehicle signals can be used by a model to make an accurate prediction of the passenger load in a vehicle. This has been broken down into several research objectives. First, we presented a data engineering pipeline to extract the features from several data sources: AFC data, GTFS data and vehicle sensor data and we have used this pipeline to construct a comprehensive dataset. Using the extracted data, we have conducted a feature analysis to explore potential relationships between the features and the passenger load or the passenger flow. These insights were used to formulate a model design where we have proposed two real-time prediction models: the PLP and LFF model. Moreover, a Seasonal ARIMA-based forecasting model has been designed along with several variations to make short-term forecasts of the passenger load using real-time signals. Both the real-time and forecasting models have been empirically evaluated. Based on the results, the LFF model has been interpreted using a SHAP value analysis in order to validate results from the feature analysis. In addition, an ablation study has been performed on the LFF model to evaluate the contribution of vehicle-related features with respect to historical features.

The importance of historical features for predicting the passenger load or flow has been identified in earlier works as well [36, 38, 39], but a comparison to vehicle-related features has not yet been made. In fact, many of the vehicle-related features considered in this work are novel features with respect to the state-of-the-art literature. The feature analysis of Chapter 5 has demonstrated that almost all vehicle-related features do not show a clear and strong relationship to the passenger load or flow. The only exception is the weight estimate, which seems to have the strongest relationship to the passenger load. The quantitative analysis showed that no feature can directly predict the passenger load or passenger flow sufficiently. However, there still seems to be a group of features that have both significant linear and non-linear relationships to the passenger load and flow. The most notable of these features are the weight estimate, the HVAC temperature deltas, the door cycle count, total door open time and the outside temperature. Historical features, based on the average passenger load or flow on similar trips, have been shown to have a strong relationship to the passenger load and flow. However, for rare instances of a high passenger load or passenger flow, these features tend to underestimate the actual passenger load or flow.

The real-time models that were implemented were based on gradient boosting machines. The PLP model directly predicts the passenger load using all features as input while the LFF model predicts the passenger load in two stages using a load component and a flow component in a time-series manner. Both models were compared to a random forest baseline similar to baselines used in the related literature. The results show that both models are more accurate than the baseline to a large degree. Where the baseline achieves an RMSE score of 18.39, the LFF model achieves an RMSE of 11.60 and the PLP an RMSE of 11.63. That is a decrease of 37% for both models. Similar performance increases are

achieved in terms of the MAE and R^2 metrics as well. The LFF model is slightly more accurate than the PLP model over all of the evaluation metrics except R^2, but the differences fall within the standard deviations. The evaluation of crowding indicators shows similar results, where the baseline is surpassed in performance significantly and the results for the PLP and LFF model are close, but the LFF model is slightly better in general. The weighted F1 score of the LFF model is 0.828 while the PLP model achieves an F1 score of 0.823. For crowding indicators, the performance notably decreases for higher bins. This is most likely due to the class imbalance in the data where higher passenger load instances are increasingly rare. A similar pattern is apparent in the regression evaluation. The error analysis showed there were slight differences in performance based on the line number and the trip progress. Overall, the results indicate that the LFF model, which leverages the problem structure and incorporates some domain knowledge, is suitable for this type of problem. The LFF model's fusion component can intelligently aggregate the predictions of the load and flow components by taking the context into account. Fusion models have similarly been successfully applied previously in the literature [23, 30, 73, 74]. However, they have not yet been used to combine an ensemble of gradient boosting models. Moreover, the fact that the LFF model successfully exploits the problem structure and domain knowledge to produce a more accurate model than a naive model such as the PLP model is also in line with findings by Pasini et al. [31] and Liu et al. [33].

The potential of using the LFF model for real-time passenger load estimation has been demonstrated by the proposal of a dashboard that shows the passenger load throughout the network for all active trips. These insights may be useful for public transport operators that require an overview of the demand on the public transport network or that may need to make ad hoc and data-driven decisions that impact travellers. An evaluation using a small sample of real-time AFC data indicated that the LFF model's performance deteriorated in a real-time setting, particularly with respect to the R^2 metric. An analysis of the data had led us to suspect that this might be due to a pandemic-related effect due to the outbreak of COVID-19 during the period of the original training data. The existence of such an effect has been investigated and demonstrated by Jenelius and Cebecauer [3]. However, further research is required to demonstrate the presence of such an effect in the current context.

As the LFF model is the most accurate real-time model, it has been used for further analysis. The SHAP analysis revealed that the model relies heavily on historical features. For the load component, the weight estimate plays a significant role as well. Surprisingly, some of the features expected to have high importance in the model, based on the feature analysis, did not seem to be as important in the post hoc analysis. These are features such as the door cycle count and the HVAC temperature deltas. On the other hand, some features which were not expected to play a large role did end up playing a significant role. These are temporal features such as the time of day and day of the month but also the dwell time. The SHAP analysis also revealed that some features have an inverse relationship to the passenger load such as the historical daily passenger load. We conclude that there are some discrepancies between the results of the feature analysis and the SHAP analysis. However, both have demonstrated that the historical features are crucial and that the vehicle-related features may provide a contribution, but only if they are combined.

We have performed an ablation study on the LFF model, investigating the contribution of the vehicle-related features and the historical AFC features. The results indicate that the LFF model predicts more accurately using only vehicle-related features than when using only historical features. Excluding the vehicle-related features resulted in a performance decrease of 42% RMSE while excluding the historical features resulted in a performance decrease of only 16% RMSE. This is contrary to the results of the SHAP value analysis, which showed a strong reliance on historical features. It indicates that the model is better able to exploit the vehicle-related features in the absence of historical features. Note, however, that the model performs best when it is provided with all input features as in the original experimental setup. Similar findings have been made by Shiao et al. [35], who indicated that providing more features to an ensemble model provides better performance. Under the assumptions of the model and experiment, it seems that vehicle-related features are more powerful than historical passenger load and flow features for predicting the passenger load. This implies that in the absence of historical passenger load data, or in a data-poor environment, real-time vehicle sensor data can effectively be used to construct a reasonable estimate of the passenger load.

We have also evaluated the proposed Seasonal ARIMA-based forecasting model and several variations that incorporate GARCH and exogenous variables into the model. The results show that the Seasonal ARIMA model including both GARCH and exogenous variables is optimal. However, GARCH

only contributes a marginal amount of performance to the model. This is contrary to the findings by Ding et al. [38] and Chen et al. [25]. However, we found that this is most likely due to a lack of volatility in the data which is in line with earlier findings by Chen et al. [26]. By updating on incoming real-time signals, the models are all able to significantly improve the performance to near-perfect predictions towards the end of the trip. However, these improvements suffer from diminishing returns as the models are updated. This is most likely due to the performance already achieving an optimum. Similarly, the look-ahead error increases over larger horizons but by increasingly less significant amounts. A similar finding was made by Gallo et al. [37]. Overall, the forecasting models based on Seasonal ARIMA have been demonstrated to be suitable candidates to complement a real-time model such as the LFF model for making reliable short-term forecasts of the passenger load.

Finally, we evaluated how the forecasting models would perform in a practical setting, without access to real-time ground-truth labels. The LFF model has been used to provide real-time estimates to the forecasting models as update signals. Remarkably, the models' performance did not deteriorate significantly. Each model had its MAE score only increased by roughly 1.18, despite the LFF predictions having an MAE of 5.27. The models, overall, continually improve their forecasts when updating through these signals. However, it may occur that these updates also cause a decrease in performance as the improvement is not guaranteed to be monotonic. In this case, the two best-performing models are the model that incorporates exogenous variables and the model incorporating both GARCH and exogenous variables. Both these models outperform the baseline in this practical setting. We conclude that this demonstrates that the practical applicability of the forecasting models is strong.

8.2. Answers to Research Questions

The primary research question of the thesis, as formulated in Section 1.2, is as follows:

> How accurately can a model based on real-time tram vehicle sensor data and historical passenger flow create a real-time estimate of the passenger load in a tram vehicle?

Compared to other models, such as the baseline, having real-time vehicle sensors clearly offers an advantage for a model that leverages those. Complementing these real-time vehicle-related features with historical passenger load and flow features provides an even better basis for prediction. We can confidently answer that the proposed LFF model can make accurate passenger load predictions, with a mean absolute error of 7.83, which may vary depending on the instance under consideration. That is a significant improvement upon the baseline model used for comparison, which has a mean absolute error of 12.49. In the following paragraphs, we will provide a detailed but concise answer to each of the sub-questions as formulated in the introduction.

RQ1: What features of the available real-time vehicle data are most descriptive of or correlated with the passenger load? The feature analysis and SHAP value analysis have shown that features may have varying relationships to the passenger load, depending on the metric or the model under consideration. It is clear that the weight estimate was the most powerful vehicle-related feature to predict the passenger load. However, it was also found in the feature analysis that features related to the doors and HVAC units in the vehicle have a relatively strong relationship to both the passenger load and flow as well. On the other hand, the SHAP value analysis of the LFF model also revealed that temporal factors such as the time of day play a large role in the prediction of the passenger load as well, mainly in the load and fusion components. Hence, we conclude that the HVAC and door features, along with the weight estimate are the most strongly related features from the vehicle data with respect to the passenger load. However, features such as the historical passenger load or temporal features have a strong relationship as well.

RQ2: What is the best method, in terms of estimation accuracy, to make real-time passenger load estimations? The proposed LFF model, which leverages the problem structure by predicting the passenger load and flow separately and making predictions in a time-series manner, provides the best estimation accuracy. A 5-fold cross-validation experiment revealed an RMSE score as low as 11.60, an MAE score as low as 7.83 and a R^2 score of 0.844. The evaluation of crowding indicators showed that the model could achieve a weighted F1 score as high as 0.823. The underlying models are gradient boosting machines which were found to provide the best performance for the current scenario.

The LFF model outperformed the baseline for each of the metrics to a large degree. The alternative PLP model was more competitive than the baseline. However, the LFF model outperformed the PLP model across all metrics except for the R^2 metric, for which the scores were equal.

RQ3: What effect does excluding the historical passenger load data have on the accuracy of the estimating model? The performance decrease resulting from excluding the historical features is roughly 16%, in terms of RMSE. However, when considering which feature set to include, it seems that the vehicle-related features are preferable to the historical features. The LFF model incorporating the historical features rather than the vehicle-related features has an 18% lower RMSE score than the other model incorporating only the vehicle-related features. The vehicle-related features may thus provide a reasonable alternative to historical passenger load and flow features, as is demonstrated by the ablation study. In the current context and under the assumptions of the model, we can even conclude that the vehicle-related features are preferable to historical features.

RQ4: How can the real-time data be incorporated into a new model to make short-term passenger load predictions for the remainder of a vehicle trip? In order to facilitate forecasting using real-time signals, the output of the real-time model is passed to a forecasting model, which can consequently be fine-tuned using the real-time predictions. In this way, the real-time signals can be provided to a forecasting model without explicitly using them as input data. The real-time model thus acts as an intermediary between the real-time signals and the forecasting model. Our experiment has shown that these real-time signals effectively improve the score increasingly up to almost negligible deviations. When using model predictions as real-time signals, the performance is slightly impacted but still satisfactory, as the MAE is only increased by 1.18 on average. With regards to the improvements by means of updating, the same trend of improvements can be found compared to using the ground-truth labels. However, the degree of improvement is smaller.

RQ5: What is the best method, in terms of prediction accuracy, to make short-term passenger load predictions using real-time vehicle data and passenger load estimations as well as historical passenger flow records? The proposed forecasting model, Seasonal ARIMA and GARCH using exogenous variables, outperforms both the other model variations and the same random forest baseline. The performance is particularly accurate when ground-truth real-time signals are provided to the model such that it can update throughout the trip. As the model is updated, the improvements suffer from diminishing returns. The overall MAE score was 5.16 with an average improvement in error of -0.876 per signal. However, towards the end of the trip, the MAE can be as low as 1.2.

8.3. Summary of Scientific Contributions

In this section, we present a summary of the scientific contributions that are provided by the results of this thesis. The research methodology has aimed to address some of the research gaps that were identified in Section 2.1.3 and contribute to the state of the art by introducing novel insights. Here, we briefly summarise the scientific contributions:

- We have introduced a novel methodology for predicting the passenger load of public transport vehicles in real-time that uses a combination of vehicle-related features and the analysis of their relation to the passenger load and flow that have previously not been considered in research.
- We have further expanded on this analysis by analysing the relevance of these features compared to historical passenger load and flow data in an ablation study, showing that vehicle-related features are a reasonable alternative to historical features, which may not be available in a data-poor environment.
- We have proposed a novel LFF model, which exploits the problem structure and domain knowledge to make accurate passenger load predictions in a time-series manner.
- We have demonstrated how real-time passenger load signals can be incorporated into a forecasting model to make accurate short-term passenger load forecasts for the remainder of a trip.

- We have introduced a novel forecasting method in the current context, regression with SARIMA and GARCH errors, that updates and fine-tunes based on incoming signals, even if these signals are real-time estimations of the passenger load. Notably, these forecasts are made from the perspective of the vehicle rather than the station.
- We have demonstrated the effect of incoming real-time signals on the accuracy of the forecasting model and how improvements change over different forecasting horizons.

8.4. Implications

In this work, we have introduced several contributions to complement the scientific literature on passenger load prediction in terms of practical implications. These contributions may have notable implications in a practical sense. In this section, we discuss some of these implications.

Section 1.1 has made the relevance of the current problem clear. The introduced LFF model offers accurate and interpretable passenger load predictions in real-time. By deploying this model in practice, transport operators and policymakers may be able to make data-driven decisions in order to optimally manage their respective traffic networks. In turn, this may reduce peak load and congestion in the urban transport network and reduce wasteful carbon emissions. Moreover, the available rolling stock can then be utilised more efficiently by ensuring that the vehicles are scheduled using these data-driven insights. This does not only benefit the transport operator but the commuter as well. An example of such usage has been provided by the proof-of-concept dashboard in Section 7.4.

With the analysis of various vehicle-related features, which have to the knowledge of the author not been applied in this context before, operators are able to decide which sensors in the vehicle are relevant to install and extract data from such that these can be used for feature extraction. The results indicate that the LFF model, using data from pre-existing sensors, can be a feasible alternative to expensive passenger load measurement equipment such as infrared cameras. The results may serve as a basis for further analysis of vehicle-related features, expanding the domain to focus more on such features rather than patterns in passenger load data

The proposed forecasting model based on exogenous variables, a novel method of forecasting passenger loads, may further assist transport operators to acquire insights into the expected load in the short term. By updating the forecasting model with real-time signals, possibly provided by the LFF model, forecasts will become increasingly reliable and could lead to further data-driven decision making. Moreover, commuters may be provided with an indication of the expected passenger load on their commute and may take that into account while planning their itinerary.

We discuss implications for a real-time product of the proposed models in additional detail, based on the real-time dashboard proposed in Section 7.4. Practical use cases for such a real-time implementation of the LFF model vary but can be defined in three categories. The first is ad hoc decision making based on real-time data. Unexpected peaks in public transport demand can be observed and addressed by allocating additional vehicles or alternative means of transportation. In addition, stewards or crowd-management personnel can be directed towards “hot spots” in the public transport network. Another aspect of this ad hoc decision making is using real-time insights to respond to disruptions or vehicle malfunctions. For non-critical disruptions or malfunctions, one could respond differently based on the observed load on the network. Moreover, an operator is able to quantify the impact that certain (scheduled) disruptions could have on passengers. These insights can be improved by introducing predictions by the forecasting model. This allows for proactive decision making.

Another category is traveller information. For short-term planned trips, an indication of expected crowdedness in the vehicle can already be given. This information could be provided by the real-time LFF model for vehicles that have already started the trip on the planned itinerary, but also by the forecasting model for more accurate estimations. Besides providing individualised information to travellers, crowding indicators can also be provided at a stop level. At the station, signs could be used to indicate the passenger load of the arriving vehicle. In cases of very high levels of crowdedness, suggestions for alternative routes could be provided to the passengers.

Finally, the insights of the real-time model can be used for post hoc analyses. For instance, one of these can be an evaluation of the timetable, considering whether additional trips need to be scheduled in peak periods. In addition, specific insights about the utilisation of certain vehicles and stations can be acquired as well. Vehicles that have frequently executed high-load trips could be more susceptible to wear and tear to the parts and may thus require additional maintenance.

8.5. Limitations

During the course of the thesis, we have identified several limitations of the work. We provide an overview of these limitations and, where appropriate, provide an explanation for their presence or suggestions for future improvement.

First of all, the size of the evaluation dataset was limited due to errors in the data engineering process. Out of the 15,000 sampled trips, data for only 12,204 trips could be collected. These errors could be due to faults in the engineering pipeline, but also due to missing or erroneous data in the data sources. Several situations which may lead to faults are described in Section 4.4. While the size of the dataset was still sufficient for the experiments in this work, the relatively large amount of errors indicates that the data engineering pipeline could be improved.

A related limitation is given by the calculation of the passenger load from the AFC data. Due to erroneous data in the AFC data collection, the calculated passenger load could be different from what would have been the actual passenger load for those instances. The applied method has been optimised to minimise errors in the calculation, as discussed in Section 4.2.2. However, as errors still remained in the method, a 100% accuracy of the calculated passenger load values cannot be guaranteed. As these values are used as ground-truth labels in this work, this may have had consequences for the results. However, we assume that the impact of these potential errors is limited and would have had a negligible effect.

The feature analysis has focused on the total flow rather than the board and alight counts. While the total flow is a summary measure, making the analysis more convenient, this is not what a model such as the flow component of the LFF model would be predicting. The choice of this target variable rather than the two board and alight counts are based on the assumptions that these are highly related and would exhibit the same relationships to the features. However, different outcomes could have resulted if the feature analysis was based on comparing the features to the board and alight counts separately. Another reason to perform the feature analysis this way was that the LFF model's flow component would initially only predict the total flow.

Another limitation is that the selection of features for the components of the load and flow components is based on the results of the feature analysis. However, the SHAP value analysis of the LFF model has shown that some of the features play a less significant role than some other features contrary to the results of the feature analysis. Hence, this may indicate that other sets of features could have been experimented with to yield a more accurate model. On the other hand, the SHAP value analysis is a post hoc analysis of the model and the conclusions are therefore difficult to generalise outside the assumptions of the model.

The LFF model uses a data split of 0.8 and 0.2 to train the load and flow components and the fusion component. This split ratio has been defined by means of trial-and-error and was based on the notion that the load and flow models are relatively large models, in terms of the number of input features and estimators, and require more training data to have a high goodness-of-fit. However, using different data splits could possibly have yielded better results. Moreover, the hyperparameters of the fusion model have not been tuned due to the complexity of performing a grid search on the whole LFF model. A grid search for optimal parameters for the fusion component could thus have yielded a more optimal overall model.

The forecasting model uses a GARCH (1,1) model. This is often the standard configuration for GARCH in the literature [67]. However, different configurations could have been experimented with but weren't due to the insignificant contribution by GARCH. If the GARCH model would have contributed more significantly to the results, different configurations would have been considered. However, it appeared that there was no high volatility in the time series. The literature has successfully applied GARCH models using different parameters in passenger flow forecasting scenarios [20]. However, GARCH seems to only contribute in cases where there is significant volatility in the time series which did not seem to be the case in the current work.

Finally, the experimental evaluation of the forecasting model has been performed using data from only one line. It could have been the case that the results would have been significantly different on other lines. The current experiment serves as a proof of concept, such that the feasibility of forecasting short-term passenger loads using Seasonal ARIMA-based models can be demonstrated. However, it is uncertain whether the current results are demonstrative of other routes in the current context or whether the results would have been significantly different in other scenarios. Only an exhaustive evaluation of all lines could demonstrate whether the evaluated line is exemplary for the other lines.

8.6. Future Work

The work presented in this thesis is by no means an exhaustive exploration of the topics covered by this research and there are certainly many avenues for possible future work. Moreover, the results presented and summarised in this chapter may also serve as a basis for exploring novel research questions. Therefore, in this section, we present recommendations for future work that may expand upon the current research and state-of-the-art literature.

The foremost recommendation is to experiment with the extraction of additional features. The acquired vehicle sensor data offers a wide variety of information. However, for many features, it was difficult to extract an aggregated feature based on the sensor signals. An example of such a feature is the consumed electricity from the overhead powerline. It could potentially provide a good indication of the passenger load, but could not be aggregated into a feature. Another example is related to the stop buttons in the vehicle which are pressed to signal the driver that passengers wish to alight the vehicle at the next stop. These signals were available, but could not be identified in the data due to a lack of documentation. However, these signals may provide additional information to the prediction models. Moreover, the inclusion of additional external features may contribute to the estimation accuracy of the models as well. Factors such as the weather forecast [75] or public holidays [36] could be particularly useful for the pipeline. In the context of an urban environment, the presence of large-scale public events such as sports matches in a nearby stadium could be included as well. In the current research, this factor was not taken into account due to the regulations against the COVID-19 pandemic prohibiting large-scale public events. Also capturing data from other modes of transport, such as the historical load at a nearby train station or bus stop could be included, such as in the work by Ding et al. [38]. Finally, incorporating some domain knowledge about the stations as a feature could be beneficial. An approach for this could be to cluster stations based on some characteristics, such as in the work by Bai et al. [74]. Spatial dependencies between and stations have been observed [37], station-related features could thus be of benefit in related work as well.

We believe that the feature analysis has demonstrated that almost all features have the potential to contribute to the model accuracy and thus adding more features will most likely have a beneficial effect.

A related recommendation is regarding the modelling of the available features. In the current work, historical features are aggregated on three temporal levels: daily, weekly and monthly. For a given instance, the historical features are acquired over other instances at the same stop and time step. However, in many other works, the historical features are also provided for previous time steps [32, 37, 39, 71]. While most of these works take a station-centric approach to predict the passenger flow, rather than a vehicle-centric approach, modelling the features at related time steps or stations could provide additional insight for a model to capitalise on.

In the current work, the real-time models and forecasting models are compared to a random forest baseline. It may be useful to compare the models to Deep Learning methods which are also popular methods in the state-of-the-art literature [17, 30, 32–34]. LSTM models, in particular, may be relevant for the problem, due to their time-series nature. Moreover, LSTM models could also be applied to the forecasting methodology and compared to the Seasonal ARIMA-based models. Due to the complexity of Deep Learning methods, a larger dataset is most likely required for accurate results, possibly spanning several additional years of data.

We have previously discussed that the GARCH (1, 1) model may not have been optimal for the current forecasting time series. Future work could further analyse the characteristics of the time series under consideration. Ding et al. [20] have performed an extensive analysis of passenger flow time series. A similar methodology can be applied to the current context.

Another interesting avenue for future work is to combine the real-time LFF model with the Seasonal ARIMA forecasting model. We have shown that using model-predicted labels to update the forecasting model has a beneficial effect. Perhaps the forecasting model could be used in a real-time setting, where it would forecast towards the current time step while updating using previous model predictions. Subsequently, the forecast could be used as a real-time prediction or aggregated with the actual real-time prediction by the model. This approach could potentially lead to better results.

Finally, the evaluation of the real-time proof-of-concept has indicated that there is a significant difference in the passenger load distribution of data from 2022 compared to the training dataset of 2020 and 2021. We have hypothesised that this could be due to a difference in pandemic-related regulations between the datasets. This effect has been discussed in the literature as well [3]. We propose that

this hypothesis could be further investigated and expanded upon by collecting a larger dataset of data during times without pandemic-related measures. As a consequence, an analysis of the distributions in the data may allow for more definitive conclusions on whether there was a so-called pandemic-related effect in the dataset. This analysis will require the researcher to identify how the pandemic regulations affect the traveller patterns and rule out other factors that may have caused a difference in the distributions regardless of the pandemic. The results of this analysis can help understand the effects that the COVID-19 pandemic has had on public transport demand. In addition, it could provide interesting avenues of research to evaluate how the performance of state-of-the-art prediction models is different under scenarios where there is an effect on public transport demand due to a pandemic. This may, in turn, provide insights into the generalisability of these models.

A

Exploratory AFC Data Analysis

To acquire a better understanding of the AFC data, an Exploratory Data Analysis (EDA) is performed. The goal is to find common patterns in the distribution of the passenger load and flow which contribute to acquiring prior knowledge for the construction of the model. We will evaluate to what extent the passenger load and flow vary under different scenarios. Temporal factors such as the time of day and day of the week are considered as well as GTFS-related factors such as the trip's progress, the line number or the station name. The dataset that is used is the AFC data related to all the trips in the dataset from the real-time model evaluation in Section 7.1.

A.1. Analysis

First, we consider the overall distribution of the passenger load, as displayed by Figure A.1. It is immediately obvious that the passenger load data is highly imbalanced and ostensibly follows an exponential distribution. A similar distribution has been observed by Gallo et al. [37]. Intuitively, this makes sense as a trip often starts and ends with a low passenger load which peaks somewhere throughout the trip. Hence, the same figure also shows the distribution of the peak passenger loads of all vehicle trips. This distribution shows a different distribution, resembling an F distribution. The fact that the passenger load is imbalanced is a factor to take into account for model training, as a model might become biased toward lower passenger loads and may thus fail to generalise or capture rare instances.

We further examine the trend of the passenger load throughout a vehicle's trip using Figure A.2. Besides the trend of the passenger load, the passenger flow trend is examined as well. The previous hypothesis regarding the passenger load peaks is confirmed, as the figure shows a clear curve of low passenger load values at the outer parts of a trip and a peak roughly in the middle. Interestingly, it appears that the board count has higher values in the first half of the trip than the second half and the alight count shows the opposite effect. The combined effect is most likely the reason for the appearance of such a curve for the trend of the passenger load.

Other factors that may affect the passenger load are temporal factors. The specific time of day or day of the week can intuitively be expected to have a significant effect. To investigate this effect, we compare the distribution of the passenger load over various times of day and days of the week in Figure A.3. The figure displays results over various time steps on weekdays and during the weekend. It is immediately obvious that there is a difference in the distribution of the passenger load across weekdays and the weekend. Most clearly, the morning and afternoon peaks are absent during the weekend. During the weekend, there seems to be a higher passenger load in the afternoon around 13:00, as well as at night around 1:00. Overall, it is evident that both the time of day as well as the day of the week has a significant impact on the passenger load.

The previous analyses have discounted the characteristics that may be inherent to the route of the vehicle. Therefore, we furthermore consider the differences in passenger load over the different vehicle routes. Figure A.4 shows the mean peak load per line. That is, in other words, the mean of the maximum values of the trips in the dataset for that given line. By aggregating the peak load values, insight into the overall demand on that line can be acquired. The figure shows that lines 2 and 9 have the highest overall demand, the average peak load on those lines is significantly larger than the other

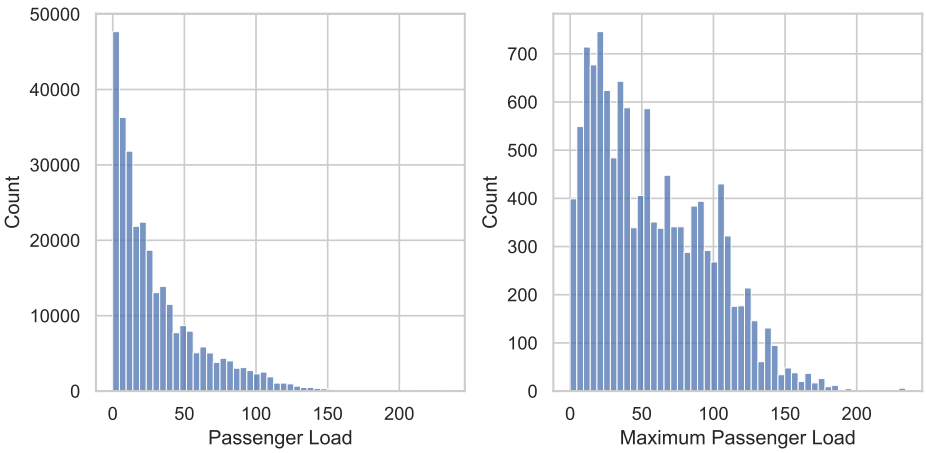

Figure A.1: Distribution of the passenger load for all instances (left) and the peak values per vehicle trip (right).

two lines. Lines 11 and 17 have the lowest overall demand.

Finally, we consider the aggregated passenger load and flow on a station level. Figure A.5 shows the distribution of the top-20 stations in terms of the passenger load and passenger flow. Overall, there seems to be less difference between individual stations in the distribution of the passenger load than the passenger flow. Also, note that the list of stations is mostly non-overlapping for the two target variables. This seems to indicate that the values of the passenger load and the passenger flow are not necessarily closely related. In essence, it indicates that stations with a high passenger flow, do not have an overall high passenger load and vice versa. Another interesting observation is that the stations with a high passenger flow are transfer stations to train stations. This indicates that these stations are part of the commute of train passengers as well. Interestingly, these stations do not seem to have a high passenger load. This indicates that there are both a large number of boarding passengers and alighting passengers at the same time. This results in the stops having similar passenger load values as for the preceding stations, despite having a high passenger flow value.

A.2. Conclusion

In the EDA we have seen that several factors are of significant influence on the passenger load. First and foremost, the distribution of the passenger load is highly imbalanced and follows an exponential distribution. This is important to take into account in designing a model and constructing a dataset, as a model might be prone to being biased towards low passenger load values. This may be addressed by oversampling trips with high peaks in passenger load values.

The analysis also showed a clear trend of the passenger load throughout a trip, where it gradually increases towards a peak in the middle and then descends back. This pattern follows intuition as well. Moreover, it seems that in the first half of the trip, most of the passenger flow is produced by boarding passengers and in the second half, the flow is mostly produced by alighting passengers.

Temporal factors appear to have a significant effect on the passenger load as well. In addition, there also seems to be an interaction between the time of day and the day of the week. For example, there does not seem to be a morning rush hour peak during the weekend days.

Finally, we observed an interesting interaction between the passenger load and the passenger flow. Self-evidently, the passenger flow affects the passenger load as can be seen in Figure A.2. On the other hand, a high passenger flow often does not indicate a high passenger load, as visible in Figure A.5. The interaction between the two target variables is ostensibly more subtle.

Overall, the EDA has shown that the passenger load is subject to various factors and exhibits certain patterns. Many of these patterns are intuitive and common in an urban transport environment: low passenger loads at the outer stations of a trip with a peak in the middle, a peak in demand during rush hour on weekdays and a large passenger flow at transit stations. A model that predicts the passenger load can exploit these patterns to derive a more accurate estimate.

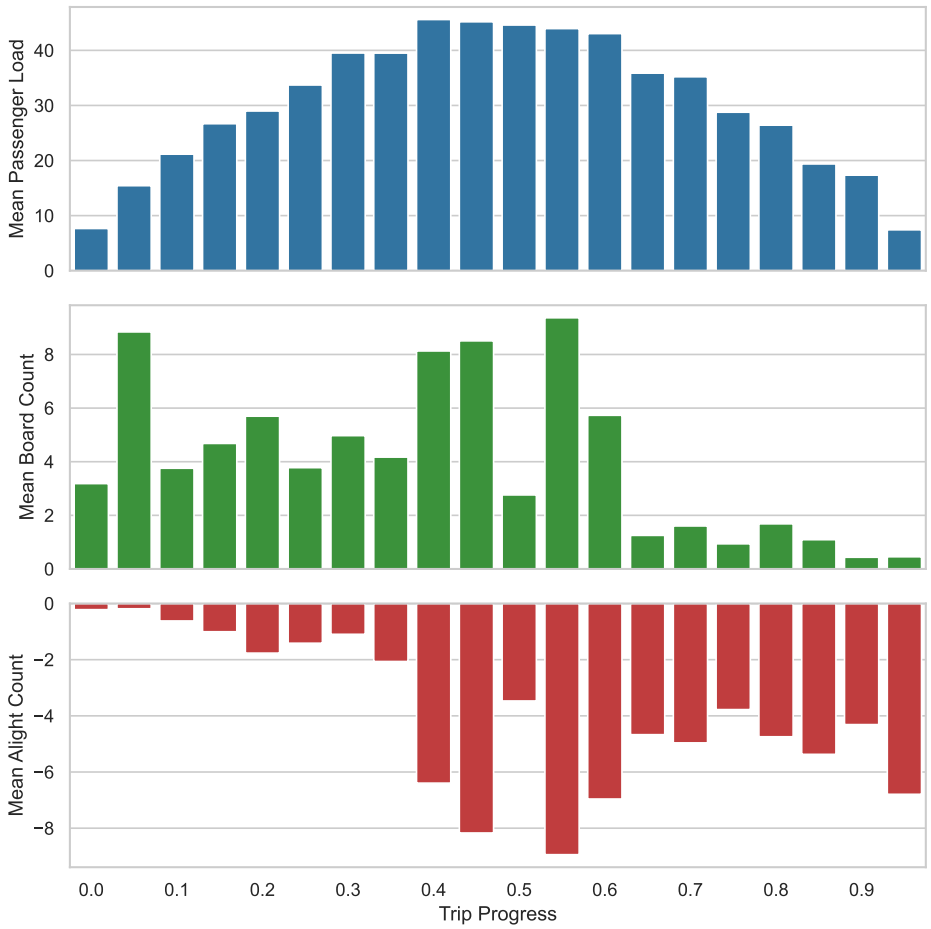

Figure A.2: Barcharts showing the trend of the mean passenger load, board count and alight count values throughout the trips. Note that the alight counts are negative as they represent the “difference” in the passenger load.

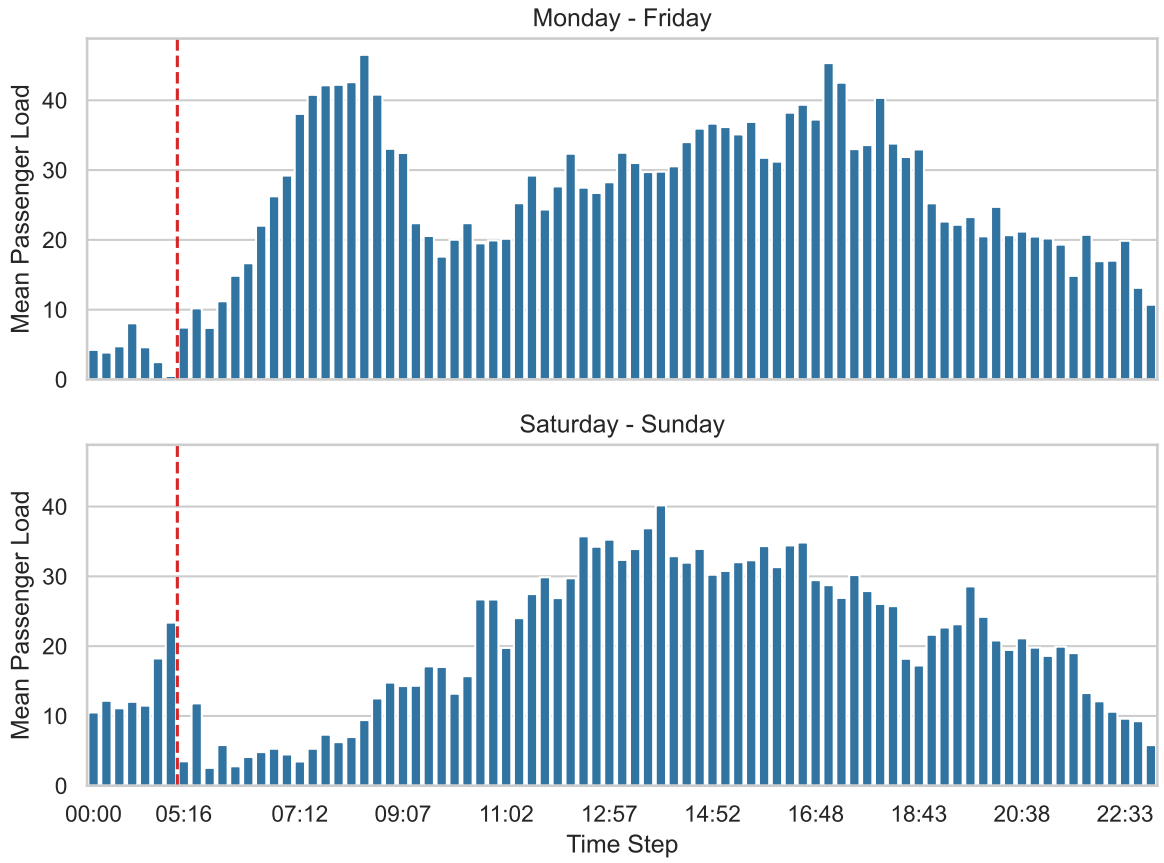

Figure A.3: Barcharts of the mean passenger load across all trips at given time steps of the day. The upper plot shows the passenger load distribution on weekdays and the lower plot shows the passenger load distribution during the weekend days. Recall that there are 100 time steps throughout the day which have a duration of approximately 15 minutes. Note that there is no public transport between roughly 1:26 and 5:16, these time steps are excluded from the figure. The gap is indicated by a vertical red line on the figure.

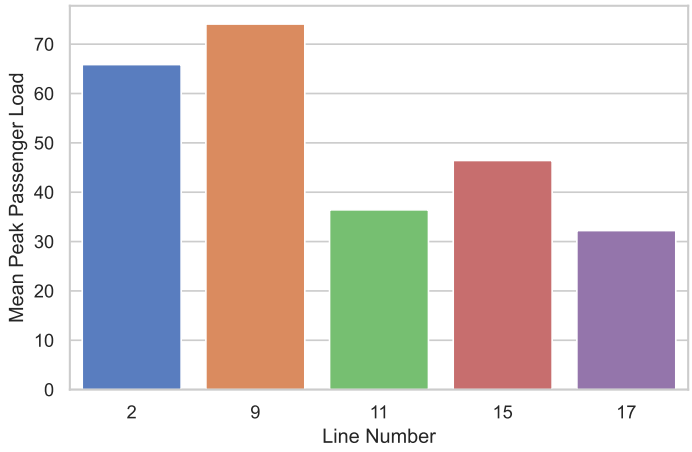

Figure A.4: Barcharts showing the mean of the peak passenger load values per trip over the varying line numbers.

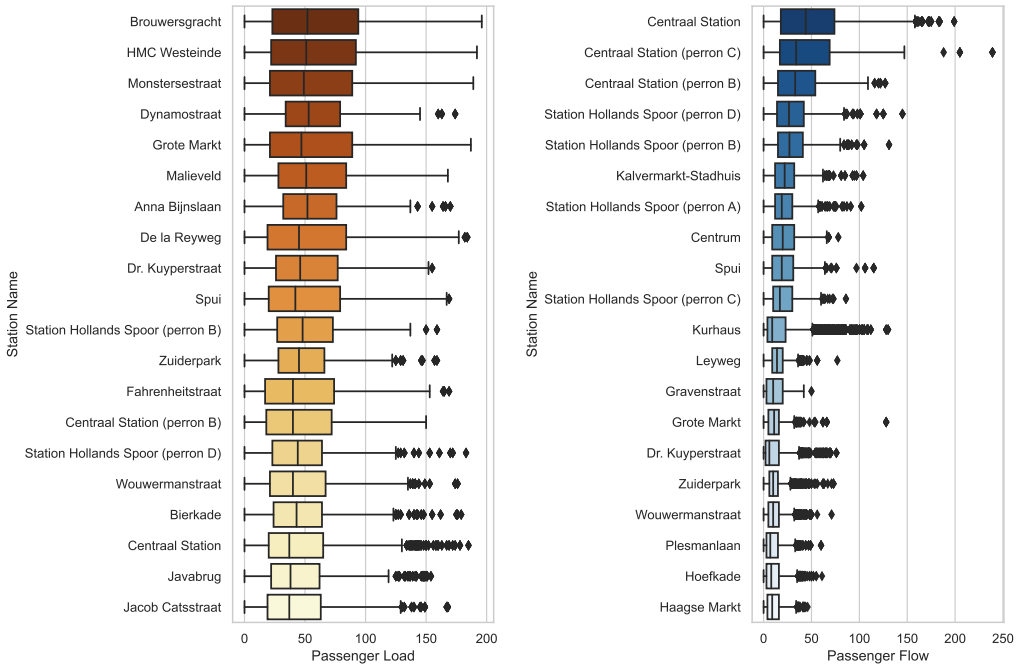

Figure A.5: Boxplots of top-20 stations ordered by mean passenger load (left) and passenger flow (right). Note that station names including “Station” are tram stations located nearby a train station.

B

Feature Overview

Table B.1 provides an overview of all features used by the models in this work. Additional details such as the data source and data value ranges are provided as well.

Table B.1: Overview of the features used by the passenger load prediction models. For each feature, the data source is noted, as well as its numerical type, the range of values it can be assigned and a brief description of its meaning. Some features are multiply defined such as the individual door-open times. In such cases, the different feature names are provided in the description. Note that the table extends to the next page.

Feature Name	Source	Type	Ranges	Description
average_daily_load	AFC	Continuous	$[0, \infty)$	Average passenger load on the same time of day
average_weekly_load	AFC	Continuous	$[0, \infty)$	Average passenger load on the same time of day and on the same day of week
average_monthly_load	AFC	Continuous	$[0, \infty)$	Average passenger load on the same time of day and on the same day of month
average_daily_board_count	AFC	Continuous	$[0, \infty)$	Average passenger board count on the same time of day
average_weekly_board_count	AFC	Continuous	$[0, \infty)$	Average passenger board count on the same time of day and on the same day of week
average_monthly_board_count	AFC	Continuous	$[0, \infty)$	Average passenger board count on the same time of day and on the same day of month
average_daily_alight_count	AFC	Continuous	$(-\infty, 0]$	Average passenger alight count on the same time of day
average_weekly_alight_count	AFC	Continuous	$(-\infty, 0]$	Average passenger alight count on the same time of day and on the same day of week
average_monthly_alight_count	AFC	Continuous	$(-\infty, 0]$	Average passenger alight count on the same time of day and on the same day of month
line_number	AFC	Nominal	2, 9, 11, 15, 17	Line number of the current trip
prev_stop_is_outer_station	AFC	Boolean	True, False	Station of last stop is a start/end station of the trip
trip_progress	AFC	Continuous	(0, 1)	Fractional progress in the trip between 0 and 1
dr_Xa_open_time	Vehicle	Discrete	$[0, \infty)$	Door-open time (s) of door Xa at last stop (X being one of 1 – 5)
dr_Xb_open_time	Vehicle	Discrete	$[0, \infty)$	Door-open time (s) of door Xb at last stop (X being one of 1 – 5)

Table B.1: Overview of the features used by the passenger load prediction models. For each feature, the data source is noted, as well as its numerical type, the range of values it can be assigned and a brief description of its meaning. Some features are multiply defined such as the individual door-open times. In such cases, the different feature names are provided in the description. Note that the table extends to the next page.

Feature Name	Source	Type	Ranges	Description
total_door_open_time	Vehicle	Discrete	[0, ∞)	Sum of door-open times at last stop
door_cycle_count	Vehicle	Discrete	[0, ∞)	Amount of door open/close cycles at last stop
prev_stop_line_headway	GTFS	Discrete	(0, ∞)	Time (s) between arrival of last vehicle of the same line and direction and current vehicle at last stop
prev_stop_overall_headway	GTFS	Discrete	(0, ∞)	Time (s) between arrival of any vehicle and current vehicle at last stop
average_out_temp	Vehicle	Continuous	(-273.15, ∞)	Average outside temperature (C) measured between departure at last stop and arrival at next stop
hvacX_mode	Vehicle	Nominal	[0, 4]	HVAC mode used longest between departure from last stop and arrival at next stop of HVAC unit X (X being one of 1, 2, 4, 5) and mode being between 0 and 4
hvacX_temp_delta	Vehicle	Continuous	(-∞,∞)	Temperature difference (C) between fresh air temperature and return air temperature of HVAC unit X (X being one of 1, 2, 4, 5)
vcu_temp_offset	Vehicle	Discrete	[-3,3]	Temperature offset configured by vehicle driver for passenger cabin
stop_time_of_day	AFC	Ordinal	[0,99]	Time step of day at last stop
day_of_week	AFC	Ordinal	[0,6]	Day of week number at last stop
day_of_month	AFC	Ordinal	[0,31]	Day of month number at last stop
vehicle_lateness	Vehicle	Discrete	(-∞,∞)	Lateness of vehicle (s) compared to scheduled arrival time
dwell_time	Vehicle	Discrete	[0,∞)	Time (s) spent stopping at last stop
weight_estimate_acc	Vehicle	Continuous	[50000,80000]	Weight estimate (kg) during acceleration from last stop
load_pred_load	LFF	Continuous	[0,∞)	Passenger load prediction by LFF Load component, provided to LFF Fusion component only
load_pred_flow	LFF	Continuous	[0,∞)	Passenger load prediction by LFF Flow component, provided to LFF Fusion component only

C

Detailed Feature Analysis Results

Tables C.1 and C.2 show the complete quantitative feature analysis results as conducted in Section 5.1. All features as shown in Table B.1 are included in the analyses. For nominal features, the value of the feature is appended to the end of the feature name.

Note that the data contains some values which are not considered in the other analyses. These were values that were unspecified or placeholder values. As these values were part of the dataset during the feature analysis, they are provided in the detailed results. The exceptions are as follows. The line number feature contains the values 42 and 59, these are values used for temporary lines and are not part of the normal timetable. The HVAC Mode features contain “nan” values, which are null values for instances where the value could not be determined.

Feature Name	F value	p value	R value
average_monthly_load	419856.66	0.00	0.77
average_weekly_load	407814.73	0.00	0.76
average_daily_load	318217.08	0.00	0.72
weight_estimate_acc	57114.37	0.00	0.40
average_weekly_board_count	29541.07	0.00	0.30
average_monthly_board_count	29174.08	0.00	0.30
average_daily_board_count	25148.12	0.00	0.28
prev_stop_is_outer_station	15244.80	0.00	-0.22
average_weekly_alight_count	11665.84	0.00	-0.20
average_monthly_alight_count	11192.53	0.00	-0.19
average_daily_alight_count	10273.68	0.00	-0.18
prev_stop_overall_headway	9919.77	0.00	-0.18
line_numer_9	8860.98	0.00	0.17
hvac1_mode_2	6378.83	0.00	-0.15
hvac2_mode_2	5278.15	0.00	-0.13
prev_stop_line_headway	4908.12	0.00	-0.13
line_numer_11	4851.34	0.00	-0.13
door_cycle_count	3625.51	0.00	0.11
dwell_time	3353.62	0.00	-0.11
hvac1_mode_3	2996.10	0.00	0.10
line_numer_17	2942.25	0.00	-0.10
hvac2_mode_3	2513.06	0.00	0.09
day_of_month	2273.10	0.00	-0.09
hvac2_temp_delta	2183.36	0.00	0.09
day_of_week	2060.64	0.00	-0.08
hvac1_temp_delta	2002.49	0.00	0.08
hvac5_temp_delta	1763.72	0.00	0.08
hvac4_temp_delta	1652.52	0.00	0.08
stop_time_of_day	1321.66	0.00	-0.07
hvac5_mode_3	1158.90	0.00	0.06
average_out_temp	1153.45	0.00	-0.06
hvac4_mode_3	943.15	0.00	0.06
dr_1a_open_time	704.79	0.00	-0.05
vehicle_lateness	671.19	0.00	0.05
hvac5_mode_1	561.36	0.00	-0.04
hvac4_mode_1	536.41	0.00	-0.04
vcu_temp_offset	468.89	0.00	0.04
line_numer_59	363.84	0.00	0.04
hvac5_mode_2	331.98	0.00	-0.03
dr_3b_open_time	239.97	0.00	0.03
line_numer_15	208.59	0.00	-0.03
hvac4_mode_2	198.92	0.00	-0.03
dr_5b_open_time	197.34	0.00	0.03
dr_1b_open_time	157.24	0.00	-0.02
dr_2b_open_time	148.42	0.00	0.02
line_numer_42	122.61	0.00	-0.02
dr_4a_open_time	115.46	0.00	-0.02
dr_4b_open_time	107.75	0.00	0.02
dr_5a_open_time	89.93	0.00	-0.02
dr_2a_open_time	60.51	0.00	-0.01
hvac2_mode_1	47.63	0.00	0.01
hvac4_mode_0	44.73	0.00	-0.01
hvac1_mode_0	39.96	0.00	-0.01
dr_3a_open_time	39.60	0.00	-0.01
trip_progress	39.08	0.00	-0.01
hvac5_mode_4	38.43	0.00	-0.01
hvac5_mode_0	37.78	0.00	-0.01
hvac4_mode_4	30.11	0.00	-0.01
hvac2_mode_0	23.74	0.00	-0.01
line_numer_2	21.88	0.00	-0.01
total_door_open_time	14.90	0.00	-0.01
hvac1_mode_1	10.65	0.00	0.01
hvac5_mode_nan	5.73	0.02	-0.00
hvac4_mode_nan	5.73	0.02	-0.00
hvac2_mode_nan	5.73	0.02	-0.00
hvac1_mode_nan	5.73	0.02	-0.00

Feature Name	MI value
average_weekly_load	1.425
weight_estimate_acc	1.356
hvac5_temp_delta	1.241
average_daily_load	1.241
hvac4_temp_delta	1.236
hvac2_temp_delta	1.203
hvac1_temp_delta	1.200
average_monthly_load	1.104
average_out_temp	1.048
average_daily_board_count	1.002
average_daily_alight_count	0.983
average_weekly_board_count	0.878
average_weekly_alight_count	0.833
average_monthly_board_count	0.456
average_monthly_alight_count	0.405
trip_progress	0.372
vehicle_lateness	0.232
total_door_open_time	0.217
dwell_time	0.209
prev_stop_overall_headway	0.185
stop_time_of_day	0.131
prev_stop_line_headway	0.113
prev_stop_is_outer_station	0.077
dr_3b_open_time	0.071
dr_4b_open_time	0.067
dr_5b_open_time	0.066
dr_1b_open_time	0.066
door_cycle_count	0.066
dr_2b_open_time	0.066
dr_3a_open_time	0.063
dr_1a_open_time	0.061
dr_5a_open_time	0.060
dr_2a_open_time	0.060
dr_4a_open_time	0.059
day_of_month	0.045
line_numer_9	0.027
day_of_week	0.018
vcu_temp_offset	0.016
hvac1_mode_2	0.014
line_numer_11	0.013
hvac1_mode_3	0.010
hvac2_mode_2	0.009
hvac2_mode_3	0.008
hvac4_mode_3	0.008
hvac4_mode_1	0.007
line_numer_17	0.007
hvac1_mode_1	0.007
hvac5_mode_3	0.005
hvac4_mode_nan	0.005
line_numer_2	0.005
hvac2_mode_1	0.005
hvac5_mode_1	0.005
hvac4_mode_2	0.004
hvac5_mode_2	0.004
line_numer_42	0.003
hvac1_mode_0	0.003
line_numer_59	0.003
hvac5_mode_4	0.002
hvac2_mode_nan	0.002
line_numer_15	0.001
hvac2_mode_0	0.001
hvac5_mode_nan	0.000
hvac4_mode_4	0.000
hvac4_mode_0	0.000
hvac1_mode_nan	0.000
hvac5_mode_0	0.000

Table C.1: Feature analysis results for all features with respect to the passenger load. The left table is showing results for the linear relationships measured in univariate regression and Pearson's correlation coefficient and the right table is showing the relationships measured in Mutual Information.

Feature Name	F value	p value	R value
average_weekly_board_count	246864.79	0.00	0.68
average_monthly_board_count	240627.33	0.00	0.67
average_weekly_alight_count	231639.61	0.00	-0.67
average_monthly_alight_count	230751.41	0.00	-0.66
average_daily_board_count	211636.06	0.00	0.65
average_daily_alight_count	208313.24	0.00	-0.65
average_weekly_load	42573.69	0.00	0.36
average_monthly_load	38611.84	0.00	0.34
average_daily_load	37036.94	0.00	0.34
door_cycle_count	18288.14	0.00	0.24
weight_estimate_acc	13743.26	0.00	0.21
prev_stop_overall_headway	12011.63	0.00	-0.20
total_door_open_time	10494.78	0.00	0.19
line_numer_9	10390.26	0.00	0.19
dr_5b_open_time	7353.72	0.00	0.16
dr_3b_open_time	6659.04	0.00	0.15
dr_4b_open_time	6571.89	0.00	0.15
dr_2b_open_time	6285.66	0.00	0.15
dr_1b_open_time	3454.05	0.00	0.11
dr_5a_open_time	3412.17	0.00	0.11
dr_2a_open_time	3064.81	0.00	0.10
dr_3a_open_time	3059.88	0.00	0.10
prev_stop_is_outer_station	2925.34	0.00	-0.10
prev_stop_line_headway	2887.91	0.00	-0.10
dr_4a_open_time	2834.75	0.00	0.10
line_numer_11	2125.47	0.00	-0.09
line_numer_17	1965.06	0.00	-0.08
hvac1_mode_2	1789.76	0.00	-0.08
dr_1a_open_time	1750.12	0.00	0.08
hvac2_mode_2	1677.13	0.00	-0.08
day_of_week	1613.42	0.00	-0.07
stop_time_of_day	1295.82	0.00	-0.07
hvac2_mode_3	1187.89	0.00	0.06
hvac1_mode_3	1134.65	0.00	0.06
hvac2_temp_delta	1128.28	0.00	0.06
line_numer_2	1032.95	0.00	-0.06
dwell_time	1031.02	0.00	0.06
hvac5_temp_delta	953.19	0.00	0.06
hvac4_temp_delta	901.30	0.00	0.06
hvac1_temp_delta	769.48	0.00	0.05
hvac5_mode_3	619.78	0.00	0.05
average_out_temp	600.40	0.00	-0.05
hvac4_mode_3	507.09	0.00	0.04
line_numer_15	424.84	0.00	-0.04
hvac5_mode_1	325.04	0.00	-0.03
vehicle_lateness	324.17	0.00	0.03
vcu_temp_offset	280.77	0.00	0.03
hvac4_mode_1	276.57	0.00	-0.03
day_of_month	183.97	0.00	-0.03
hvac5_mode_2	178.20	0.00	-0.02
hvac4_mode_2	132.93	0.00	-0.02
trip_progress	126.32	0.00	0.02
line_numer_42	86.75	0.00	-0.02
line_numer_59	54.80	0.00	0.01
hvac1_mode_0	21.51	0.00	-0.01
hvac4_mode_0	19.01	0.00	-0.01
hvac5_mode_0	18.77	0.00	-0.01
hvac2_mode_0	14.69	0.00	-0.01
hvac1_mode_1	12.40	0.00	-0.01
hvac2_mode_1	11.31	0.00	-0.01
hvac4_mode_4	4.75	0.03	-0.00
hvac5_mode_4	3.57	0.06	-0.00
hvac1_mode_nan	0.87	0.35	-0.00
hvac4_mode_nan	0.87	0.35	-0.00
hvac2_mode_nan	0.87	0.35	-0.00
hvac5_mode_nan	0.87	0.35	-0.00

Feature Name	MI value
weight_estimate_acc	0.891
hvac5_temp_delta	0.870
hvac4_temp_delta	0.867
hvac1_temp_delta	0.836
hvac2_temp_delta	0.836
average_weekly_load	0.791
average_daily_board_count	0.774
average_daily_alight_count	0.771
average_daily_load	0.738
average_out_temp	0.700
average_weekly_board_count	0.681
average_weekly_alight_count	0.676
total_door_open_time	0.499
average_monthly_load	0.498
dwell_time	0.458
average_monthly_alight_count	0.426
average_monthly_board_count	0.421
trip_progress	0.281
door_cycle_count	0.252
dr_3b_open_time	0.164
dr_5b_open_time	0.154
dr_2b_open_time	0.150
dr_4b_open_time	0.146
dr_3a_open_time	0.145
prev_stop_overall_headway	0.138
dr_5a_open_time	0.133
dr_2a_open_time	0.131
dr_4a_open_time	0.128
dr_1b_open_time	0.124
vehicle_lateness	0.121
dr_1a_open_time	0.112
stop_time_of_day	0.093
prev_stop_line_headway	0.074
line_numer_9	0.029
prev_stop_is_outer_station	0.023
day_of_month	0.022
day_of_week	0.014
hvac2_mode_3	0.008
line_numer_17	0.008
line_numer_11	0.007
vcu_temp_offset	0.006
hvac1_mode_2	0.006
hvac4_mode_3	0.006
hvac2_mode_2	0.005
line_numer_2	0.005
hvac1_mode_3	0.002
hvac4_mode_2	0.002
hvac2_mode_nan	0.002
hvac1_mode_nan	0.002
hvac1_mode_1	0.002
hvac5_mode_4	0.002
hvac4_mode_1	0.001
hvac5_mode_2	0.001
hvac5_mode_3	0.001
line_numer_15	0.001
hvac5_mode_nan	0.001
hvac5_mode_1	0.001
hvac5_mode_0	0.001
hvac4_mode_0	0.001
hvac1_mode_0	0.001
line_numer_59	0.001
hvac4_mode_nan	0.001
hvac2_mode_1	0.000
hvac4_mode_4	0.000
hvac2_mode_0	0.000
line_numer_42	0.000

Table C.2: Feature analysis results for all features with respect to the passenger flow. The left table is showing results for the linear relationships measured in univariate regression and Pearson's correlation coefficient and the right table is showing the relationships measured in Mutual Information.

D

Grid Search Results

This chapter provides details about the results of the grid search as described in Section 6.1.3. The grid search is executed for all real-time models: the PLP model, LFF model (the load and flow components) and the random forest baseline. As the PLP and LFF models both use gradient boosting machines as their base models, they are tuned to the same parameter sets. The gradient tree boosting models are tuned in two stages to reduce the exponential growth in parameter sets to evaluate: the boosting stage and the tree stage. For each stage, the related parameters are tuned while the other parameters are fixed. It is expected that the two groups of parameters do not affect each other significantly. In the following sections, the top-ten results for each of the models are shown based on their R^2 score.

Parameter Name	Boost Search Values	Tree Search Values
n_estimators	[50, 100, 150, 250, 325, 400]	100
learning_rate	[0.01, 0.05, 0.1, 0.25]	0.1
subsample	[1.0, 0.75, 0.5, 0.25, 0.1]	0.5
loss	[squared_error, absolute_error, huber]	friedman_mse
min_samples_split	2	[2, 3, 5, 10, 15, 30]
min_samples_leaf	1	[1, 3, 5, 10, 15, 30]
max_depth	3	[3, 4, 5, 7, 9, 12, 15]
max_features	`null`	[auto, sqrt, log2]

Table D.1: Gradient tree boosting model grid search values are searched in two stages, the boosting stage and the tree stage. Parameter values that are indicated by a single value are fixed during the search.

Parameter Name	Search Values
n_estimators	[25, 50, 75, 100, 150, 200, 300, 500, 700]
min_samples_split	[2, 5, 10, 20]
min_samples_leaf	1
max_depth	[`null`, 3, 5, 7, 10]
max_features	[auto, sqrt, log2, 0.5]
criterion	squared_error
min_samples_leaf	1
oob_score	false
min_impurity_decrease	0.0
bootstrap	true

Table D.2: Random forest model grid search values. Values indicated by a single value are fixed during the grid search.

D.1. PLP Model Results

Tables D.3 and D.4 show the top results of the grid search for the boost stage and tree stage for the PLP model.

n_estimators	learning_rate	subsample	loss	R2	MAE	RMSE
400	0.25	0.5	huber	0.800	9.171	13.259
400	0.25	1.0	squared_error	0.799	9.304	13.280
400	0.25	0.75	squared_error	0.798	9.281	13.320
325	0.25	1.0	squared_error	0.798	9.341	13.333
325	0.25	0.5	huber	0.797	9.207	13.344
400	0.25	0.25	huber	0.797	9.298	13.374
400	0.25	0.75	huber	0.796	9.220	13.380
250	0.25	0.5	huber	0.796	9.266	13.402
325	0.25	0.75	huber	0.795	9.261	13.428
250	0.25	1.0	squared_error	0.794	9.387	13.445

Table D.3: Boost stage grid search results for the PLP model.

min_samples_split	min_samples_leaf	max_depth	max_features	R2	MAE	RMSE
15	30	7	auto	0.807	8.784	13.026
2	30	7	auto	0.807	8.784	13.026
3	30	7	auto	0.807	8.784	13.026
30	30	7	auto	0.807	8.784	13.026
10	30	7	auto	0.807	8.784	13.026
5	30	7	auto	0.807	8.784	13.026
30	5	7	auto	0.807	8.747	13.027
15	3	9	auto	0.807	8.695	13.038
30	3	7	auto	0.806	8.777	13.050
15	3	7	auto	0.805	8.809	13.090

Table D.4: Tree stage grid search results for the PLP model.

D.2. LFF Model Results

The LFF model's load and flow components are tuned separately. The following sections show the results for each component.

D.2.1. LFF Load Component

Tables D.5 and D.6 show the top results of the grid search for the boost stage and tree stage for the LFF model's load component.

n_estimators	learning_rate	subsample	loss	R2	MAE	RMSE
400	0.25	0.75	huber	0.762	10.015	14.469
400	0.25	0.5	huber	0.761	10.021	14.508
325	0.25	0.5	huber	0.759	10.057	14.567
325	0.25	0.75	huber	0.758	10.058	14.570
400	0.1	0.5	squared_error	0.758	10.175	14.593
400	0.25	0.75	squared_error	0.757	10.183	14.603
400	0.1	0.25	huber	0.757	10.020	14.617
250	0.25	0.5	huber	0.757	10.050	14.620
325	0.25	0.75	squared_error	0.756	10.197	14.634
400	0.25	0.5	squared_error	0.756	10.248	14.643

Table D.5: Boost stage grid search results for the LFF load model.

min_samples_split	min_samples_leaf	max_depth	max_features	R2	MAE	RMSE
3	30	7	auto	0.779	9.465	13.933
5	30	7	auto	0.779	9.465	13.933
30	30	7	auto	0.779	9.465	13.933
15	30	7	auto	0.779	9.465	13.933
10	30	7	auto	0.779	9.465	13.933
2	30	7	auto	0.779	9.465	13.933
15	1	7	auto	0.779	9.501	13.952
3	1	7	auto	0.778	9.467	13.954
30	3	7	auto	0.778	9.460	13.955
3	3	7	auto	0.778	9.474	13.956

Table D.6: Tree stage grid search results for the LFF load model.

D.2.2. LFF Flow Component

Tables D.7 and D.8 show the top results of the grid search for the boost stage and tree stage for the LFF model's flow component.

n_estimators	learning_rate	subsample	loss	R2	MAE	RMSE
400	0.1	0.75	squared_error	0.804	1.758	3.488
325	0.1	0.75	squared_error	0.802	1.768	3.512
400	0.1	0.5	squared_error	0.800	1.770	3.533
400	0.1	1.0	squared_error	0.798	1.755	3.545
400	0.25	1.0	squared_error	0.797	1.739	3.550
325	0.10	0.5	squared_error	0.796	1.782	3.564
250	0.1	0.75	squared_error	0.796	1.790	3.562
325	0.25	1.0	squared_error	0.795	1.749	3.569
325	0.1	1.0	squared_error	0.794	1.769	3.572
250	0.25	1.0	squared_error	0.794	1.759	3.574

Table D.7: Boost stage grid search results for the LFF flow model.

min_samples_split	min_samples_leaf	max_depth	max_features	R2	MAE	RMSE
10	1	9	auto	0.818	1.639	3.367
3	1	9	auto	0.816	1.639	3.384
15	5	9	auto	0.816	1.633	3.383
3	5	9	auto	0.816	1.638	3.382
5	5	9	auto	0.816	1.638	3.382
10	5	9	auto	0.816	1.638	3.382
2	5	9	auto	0.816	1.638	3.382
10	1	7	auto	0.816	1.662	3.389
5	10	9	auto	0.815	1.638	3.390
3	10	9	auto	0.815	1.638	3.390

Table D.8: Tree stage grid search results for the LFF flow model.

D.3. Baseline Model Results

Table D.9 shows the top results of the grid search for the random forest baseline model.

n_estimators	max_features	max_samples	max_depth	min_samples_split	R2	MAE	RMSE
500	0.5	0.8	null	10	0.548	13.662	19.936
300	0.5	0.8	null	10	0.547	13.666	19.945
700	0.5	0.6	null	2	0.547	13.677	19.946
500	0.5	0.8	null	5	0.547	13.637	19.949
700	0.5	0.8	null	10	0.547	13.670	19.951
300	0.5	0.6	null	2	0.547	13.684	19.952
700	0.5	0.6	null	5	0.547	13.701	19.956
700	0.5	0.8	null	5	0.547	13.634	19.957
75	0.5	0.8	null	10	0.547	13.666	19.958
500	0.5	0.6	null	2	0.547	13.685	19.959

Table D.9: Grid search results for the random forest baseline model.

E

Real-Time Model Details

This chapter will describe the exact configuration of each of the real-time models. All models have been implemented using scikit-learn [41] in the Python programming language [72]. Note that the parameters for each model have been selected based on the results of the grid search. Detailed results of the grid search can be found in Appendix D. A description of the features can be found in Appendix B.

E.1. PLP Model Details

Table E.1 provides details about the final configuration of the PLP model for the experimental evaluation.

Model Class	Parameters	Features
GradientBoostingRegressor	n_estimators: 400, learning_rate: 0.25, subsample: 0.5, loss: huber, min_samples_split: 2, min_samples_leaf: 30, max_depth: 7, max_features: auto	stop_time_of_day, day_of_week, day_of_month, line_number, hvac1_mode, hvac2_mode, hvac4_mode, hvac5_mode, dr_1a_open_time, dr_2a_open_time, dr_3a_open_time, dr_4a_open_time, dr_5a_open_time, dr_1b_open_time, dr_2b_open_time, dr_3b_open_time, dr_4b_open_time, dr_5b_open_time, total_door_open_time, door_cycle_count, dwell_time, weight_estimate_acc, hvac1_temp_delta, hvac2_temp_delta, hvac4_temp_delta, average_daily_load, average_weekly_load, average_monthly_load, vehicle_lateness, hvac5_temp_delta, average_out_temp, vcu_temp_offset, prev_stop_line_headway, prev_stop_overall_headway, trip_progress, prev_stop_is_outer_station, average_daily_board_count, average_weekly_board_count, average_monthly_board_count, average_daily_alight_count, average_weekly_alight_count, average_monthly_alight_count

Table E.1: Exact model configuration for the PLP model, default parameter values have been used for parameters which have been unspecified (as per scikit-learn version 1.0).

E.2. LFF Model Details

Tables E.2, E.3 and E.4 provide details about the final configuration of each of the models in the individual components of the LFF model for the experimental evaluation. Table E.2 provides details about the load component, Table E.3 provides details about the flow component and Table E.4 provides deatils about the fusion component.

Model Class	Parameters	Features
GradientBoostingRegressor	n_estimators: 400, learning_rate: 0.25, subsample: 0.75, loss: huber, min_samples_split: 2, min_samples_leaf: 30, max_depth: 7, max_features: auto	stop_time_of_day, day_of_week, day_of_month, line_number, hvac1_mode, hvac2_mode, hvac4_mode, hvac5_mode, total_door_open_time, door_cycle_count, weight_estimate_acc, hvac1_temp_delta, hvac2_temp_delta, hvac4_temp_delta, hvac5_temp_delta, average_daily_load, average_weekly_load, average_monthly_load, average_out_temp, vcu_temp_offset, prev_stop_line_headway, prev_stop_overall_headway, trip_progress, prev_stop_is_outer_station

Table E.2: Exact model configuration for the LFF Load Component's model, default parameter values have been used for parameters that have been unspecified (as per scikit-learn version 1.0).

Model Class	Parameters	Features
MultiOutputRegressor with base estimator GradientBoostingRegressor	n_estimators: 400, learning_rate: 0.1, subsample: 0.75, loss: huber, min_samples_split: 10, min_samples_leaf: 1, max_depth: 9, max_features: auto	stop_time_of_day, day_of_week, day_of_month, line_number, dr_1a_open_time, dr_2a_open_time, dr_3a_open_time, dr_4a_open_time, dr_5a_open_time, dr_1b_open_time, dr_2b_open_time, dr_3b_open_time, dr_4b_open_time, dr_5b_open_time, total_door_open_time, door_cycle_count, dwell_time, weight_estimate_acc, hvac1_temp_delta, hvac2_temp_delta, hvac4_temp_delta, hvac5_temp_delta, vehicle_lateness, average_out_temp, vcu_temp_offset, prev_stop_line_headway, prev_stop_overall_headway, trip_progress, prev_stop_is_outer_station, average_daily_board_count, average_weekly_board_count, average_monthly_board_count, average_daily_alight_count, average_weekly_alight_count, average_monthly_alight_count

Table E.3: Exact model configuration for the LFF Flow Component's model, default parameter values have been used for parameters that have been unspecified (as per scikit-learn version 1.0).

Model Class	Parameters	Features
GradientBoostingRegressor	n_estimators: 100, loss: huber, max_depth: 7	stop_time_of_day, day_of_week, day_of_month, line_number, prev_stop_line_headway, prev_stop_overall_headway, trip_progress, prev_stop_is_outer_station, load_pred_load, load_pred_flow

Table E.4: Exact model configuration for the LFF Fusion Component's model, default parameter values have been used for parameters that have been unspecified (as per scikit-learn version 1.0)

.

E.3. Baseline Model Details

Table E.5 provides details about the final configuration of the random forest baseline model for the experimental evaluation.

Model Class	Parameters	Features
RandomForestRegressor	n_estimators: 500, max_samples: 0.8, min_samples_split: 10, max_features: 0.5	stop_time_of_day, day_of_week, day_of_month, line_number, prev_stop_line_headway, prev_stop_overall_headway, trip_progress, prev_stop_is_outer_station, average_daily_load, average_weekly_load, average_monthly_load, average_daily_board_count, average_weekly_board_count, average_monthly_board_count, average_daily_alight_count, average_weekly_alight_count, average_monthly_alight_count

Table E.5: Exact model configuration for the Random Forest Baseline model, default parameter values have been used for parameters that have been unspecified (as per scikit-learn version 1.0)

F

Ablation Study PLP

Table F.1 shows the same ablation study as performed in Section 7.3, but it is executed on the PLP model. Observe that the outcomes of the experiment are similar to the LFF model.

	PLP Complete		**PLP Vechicle**		**PLP Historical**	
	Mean	SD	Mean	SD	Mean	SD
RMSE	*11.63*	0.302	14.19	0.383	15.31	0.396
MAE	*7.87*	0.141	9.86	0.180	10.12	0.165
R2	*0.844*	0.009	0.768	0.005	0.729	0.013

Table F.1: Experimental results on evaluating the different feature sets on the PLP model. The PLP Vehicle column contains the results for the PLP model using all vehicle-related features and the PLP Historical column contains the results for the PLP model using all historical features. All results show the mean metric values with standard deviations (SD) of 5-fold cross-validation. The best mean scores are written in italics.

G

Individual Trip Forecasts

Figures G.1, G.2 and G.3 show forecasts of the passenger load on individual trips by each of the forecasting models.

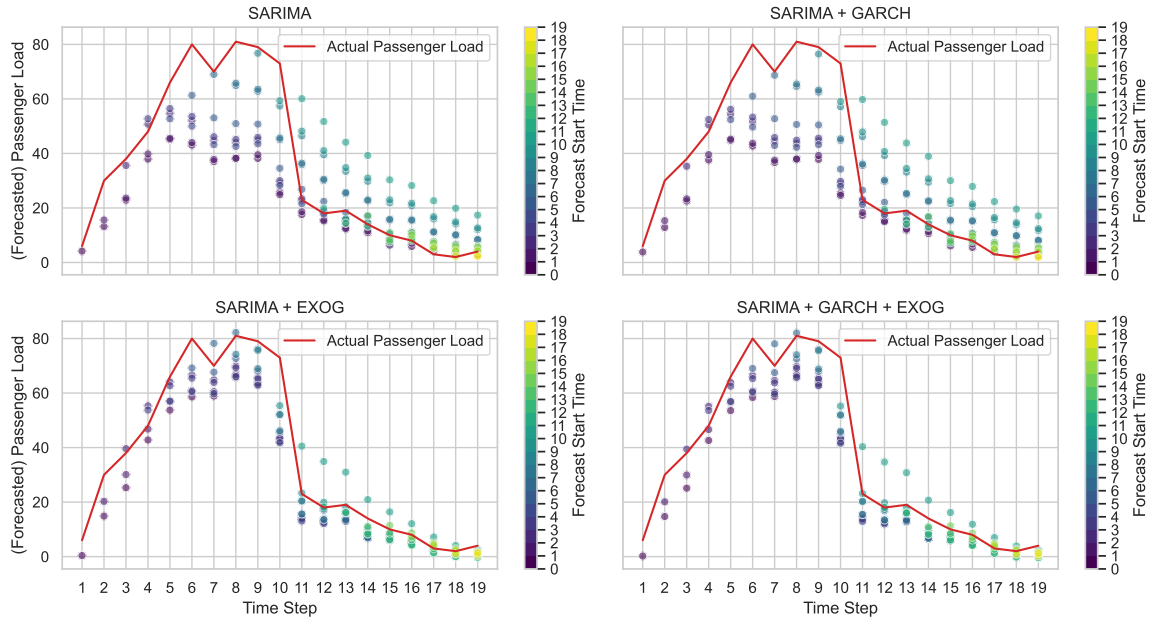

Figure G.1: Passenger load forecasts by each of the forecasting models for a trip on 2020-03-03 starting at 07:30. The figure shows the actual passenger load as a red line and individual forecasts at varying time steps using dots. As the forecasts are made at increasingly later time steps, the dots are coloured increasingly lighter accordingly.

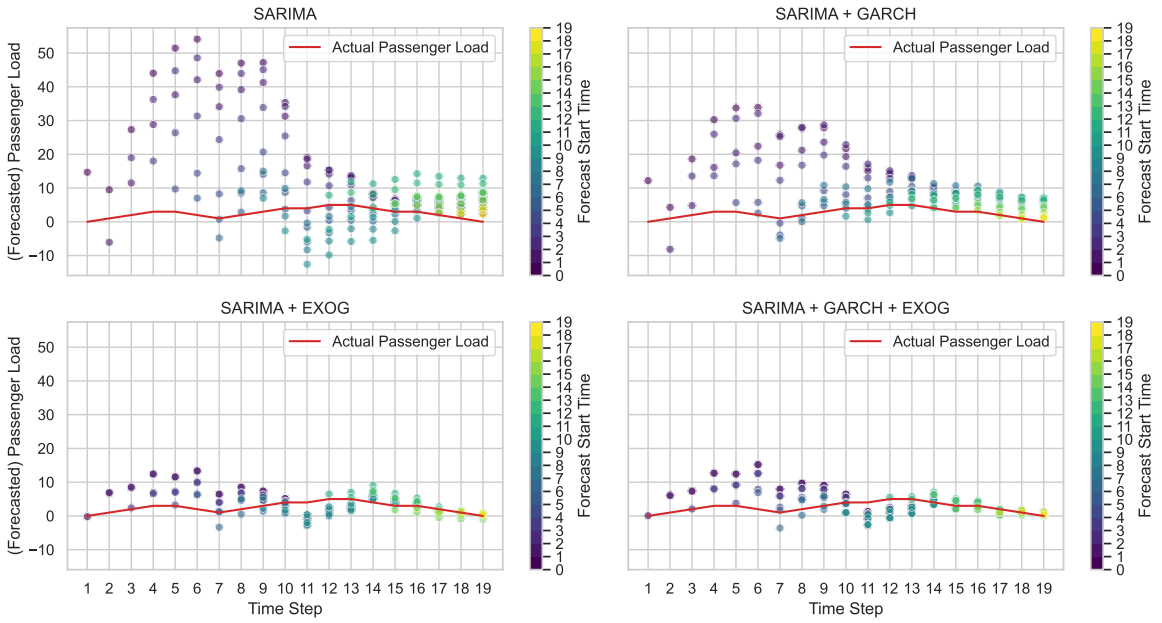

Figure G.2: Passenger load forecasts by each of the forecasting models for a trip on 2020-03-17 starting at 06:59. The figure shows the actual passenger load as a red line and individual forecasts at varying time steps using dots. As the forecasts are made at increasingly later time steps, the dots are coloured increasingly lighter accordingly.

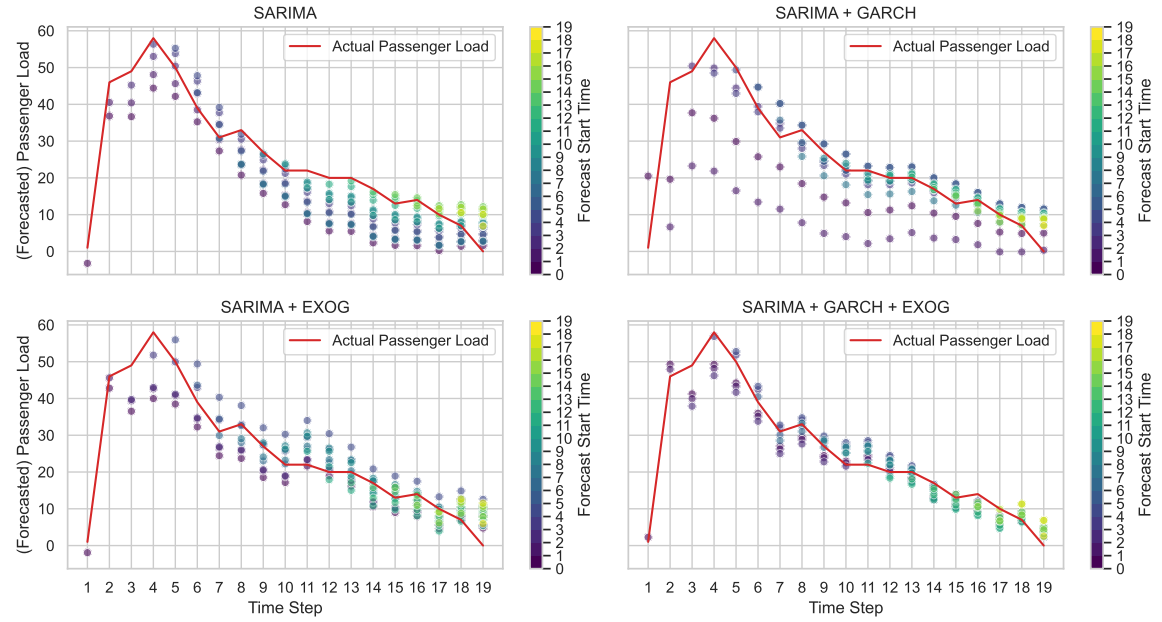

Figure G.3: Passenger load forecasts by each of the forecasting models for a trip on 2020-07-20 starting at 17:53. The figure shows the actual passenger load as a red line and individual forecasts at varying time steps using dots. As the forecasts are made at increasingly later time steps, the dots are coloured increasingly lighter accordingly.

H

Forecast Error Heatmaps

The following sections provide detailed forecasting results in terms of MAE per station and forecasting time step. Similar to the results in Table 7.8, the models have been updated using the ground-truth labels and the LFF-predicted labels. Section H.1 provides the results of forecasting updates using the ground-truth labels, which are also part of the remainder of the analysis in Section 7.5.2 and Section H.2 provides the results of forecasting updates using the LFF-predicted labels. For the reader's convenience, different colours are used in the figures depending on what type of update labels have been used.

H.1. Ground-truth Labels

Figures H.1, H.2, H.3, H.4 display heatmaps of the MAE for each model and forecasted station per forecasting time step. The results shown in these figures have been made by updating the models with the ground-truth labels of the passenger load.

H.2. LFF-Predicted Labels

Figures H.5, H.6, H.7, H.8 display heatmaps of the MAE for each model and forecasted station per forecasting time step. The results shown in these figures have been made by updating the models with the labels of the passenger load predicted by the LFF model.

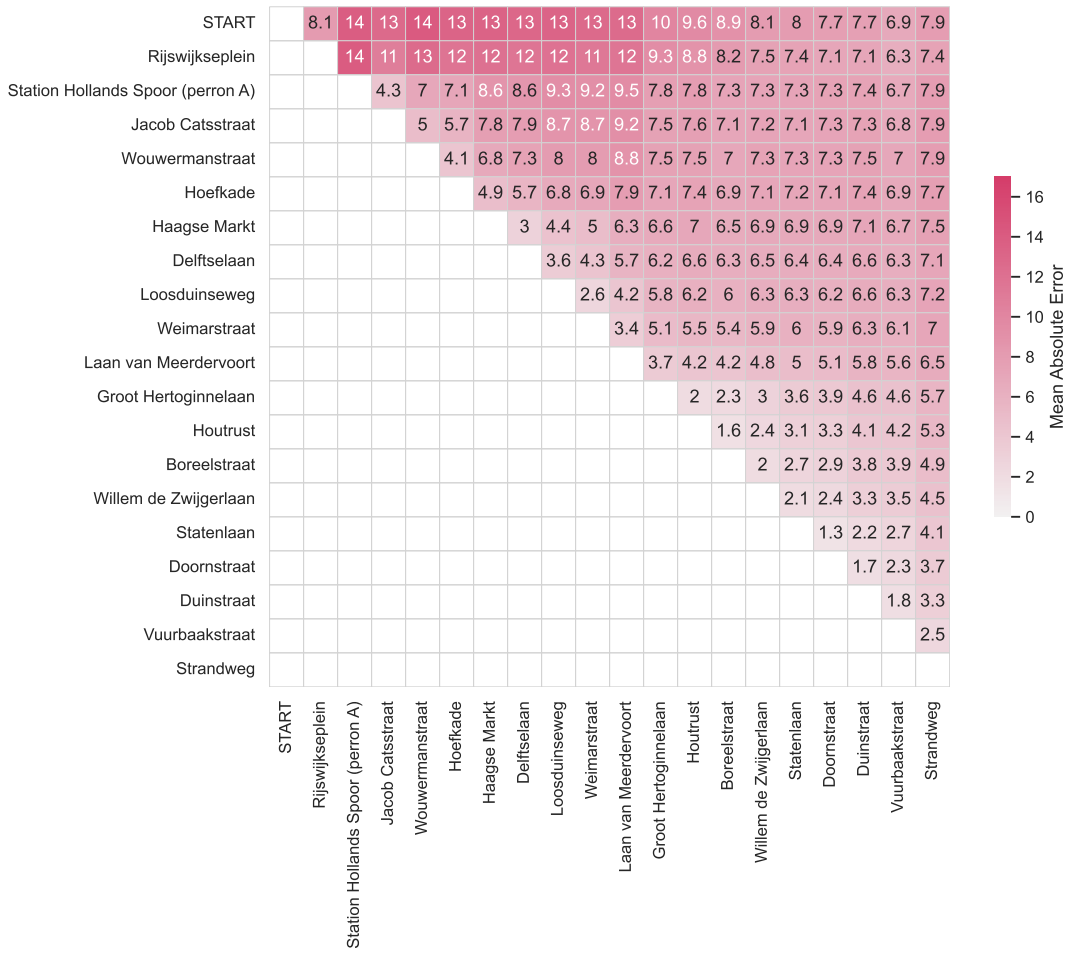

Figure H.1: Heatmap showing the MAE of the forecasts by the Seasonal ARIMA model per forecast time step. The model has been updated using ground-truth labels of the passenger load.

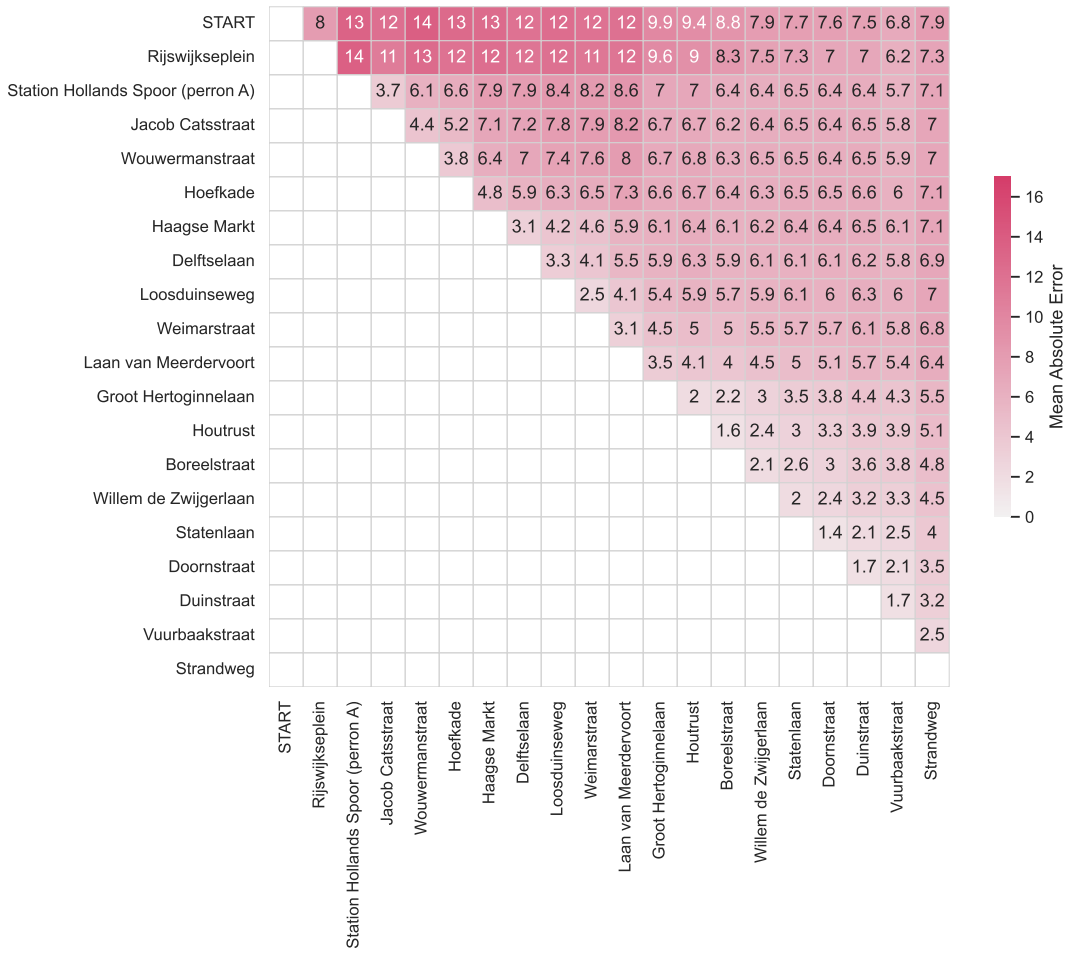

Figure H.2: Heatmap showing the MAE of the forecasts by the Seasonal ARIMA with GARCH model per station and per forecasting time step. The model has been updated using ground-truth labels of the passenger load.

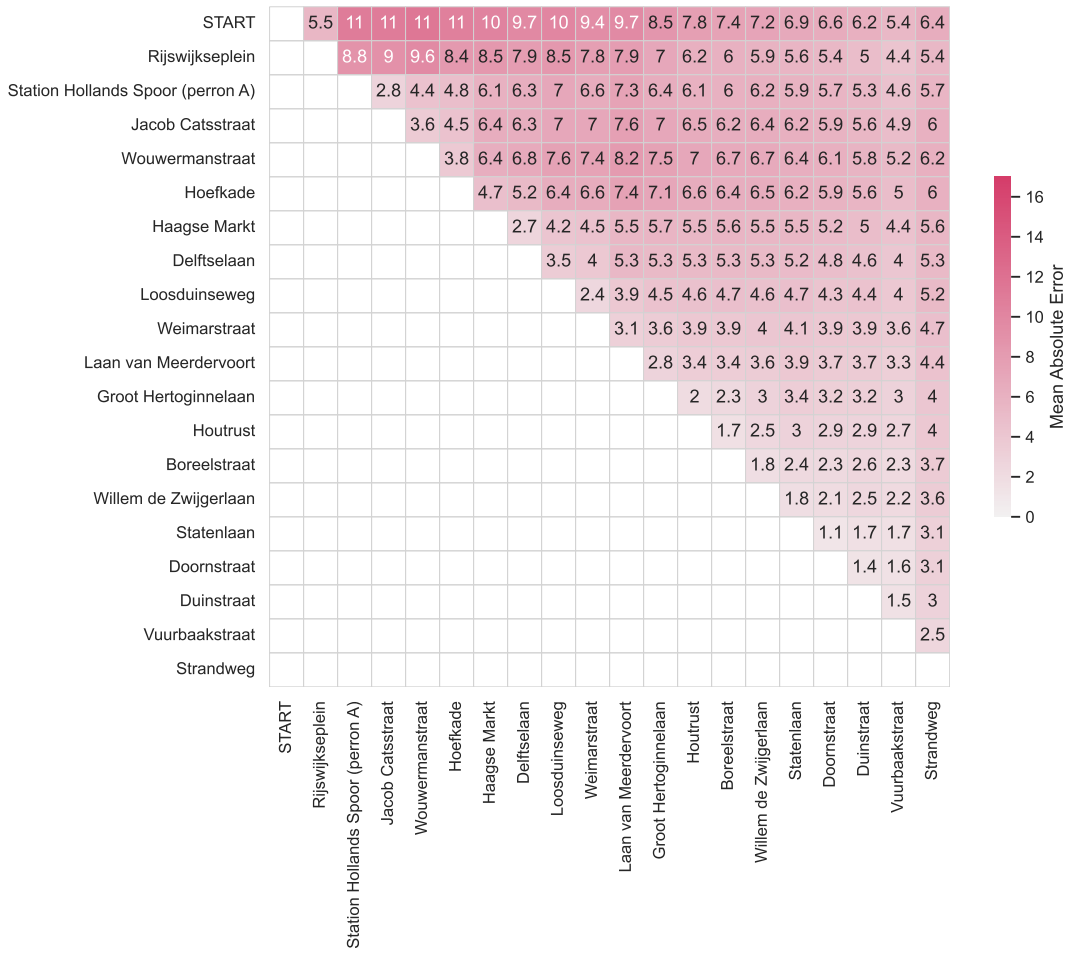

Figure H.3: Heatmap showing the MAE of the forecasts by the Seasonal ARIMA with exogenous variables model per station and per forecasting time step. The model has been updated using ground-truth labels of the passenger load.

Figure H.4: Heatmap showing the MAE of the forecasts by the Seasonal ARIMA with GARCH and exogenous variables model per station and per forecasting time step. The model has been updated using ground-truth labels of the passenger load.

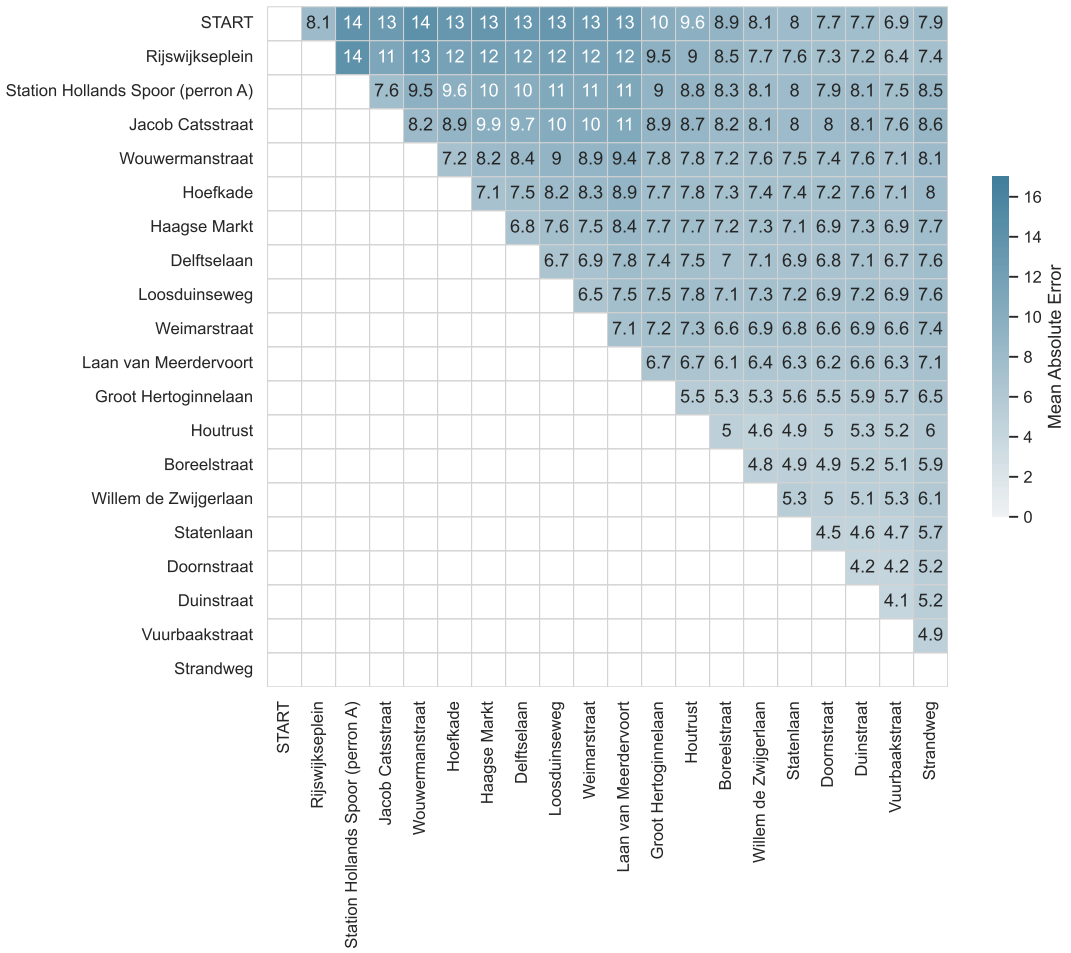

Figure H.5: Heatmap showing the MAE of the forecasts by the Seasonal ARIMA model per station and per forecasting time step. The model has been updated using labels of the passenger load predicted by the LFF model.

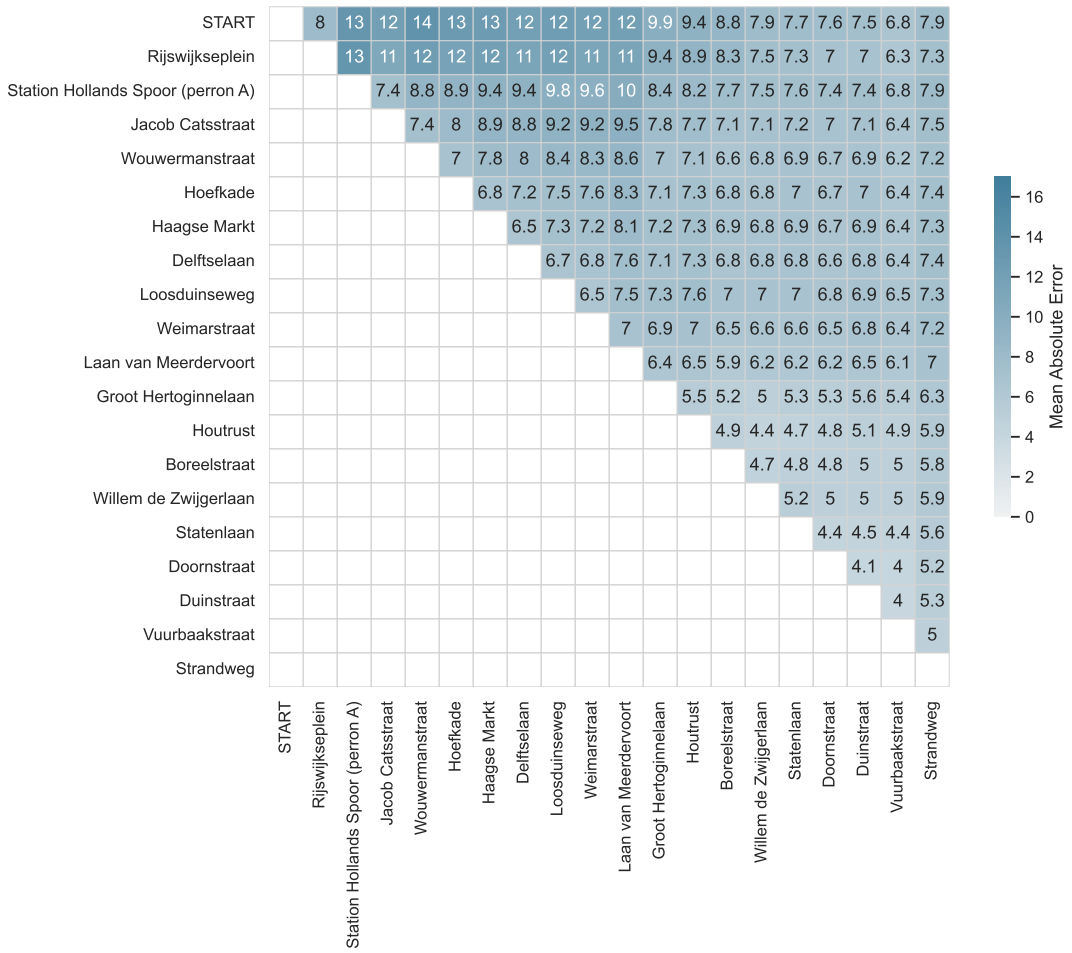

Figure H.6: Heatmap showing the MAE of the forecasts by the Seasonal ARIMA with GARCH model pper station and per forecasting time step. The model has been updated using labels of the passenger load predicted by the LFF model.

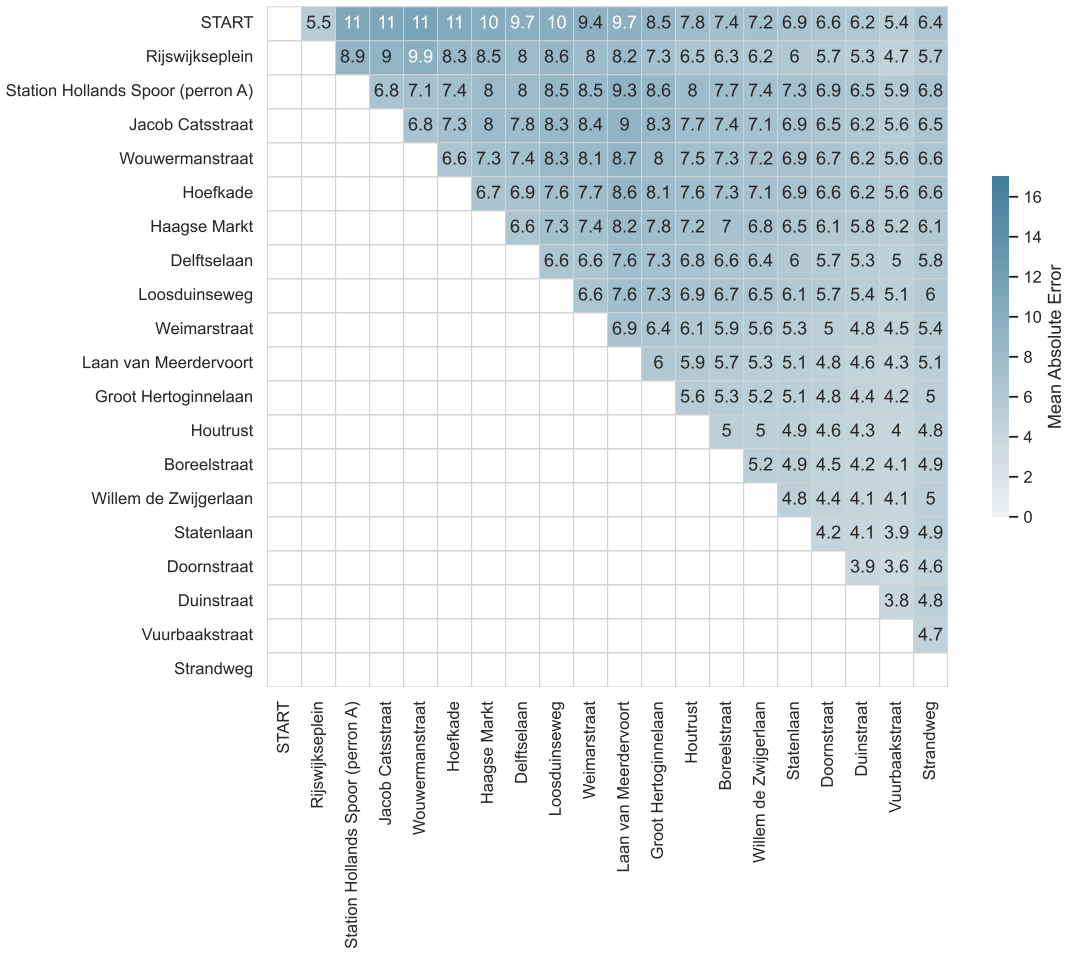

Figure H.7: Heatmap showing the MAE of the forecasts by the Seasonal ARIMA with exogenous variables model per station and per forecasting time step. The model has been updated using labels of the passenger load predicted by the LFF model.

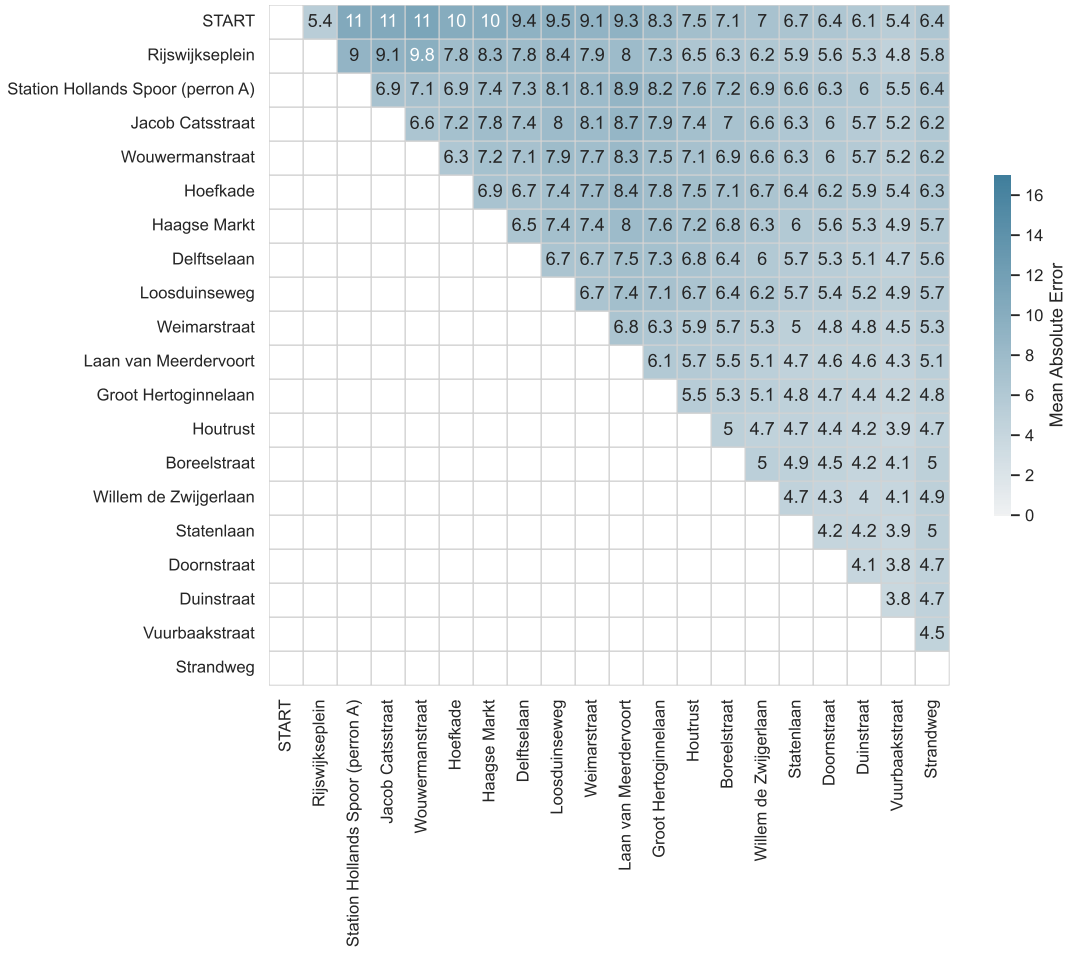

Figure H.8: Heatmap showing the MAE of the forecasts by the Seasonal ARIMA with GARCH and exogenous variables model per station and per forecasting time step. The model has been updated using labels of the passenger load predicted by the LFF model.

Forecasting Runtimes

Table I.1 shows the runtime of each of the forecasting models in the experiment. The total runtime for updated models includes the collection of history from the training dataset, the initial fitting, the prediction and the updating using incoming signals. For runs without retraining, the total runtime only includes the collection of training data, the initial fitting and the forecasting. The total runtime has been divided by the size of the test set (144 trips) to get an approximate runtime per trip. Note that the actual runtime per trip may vary, as the update procedure may take shorter or longer depending on the complexity of the maximum likelihood estimation. Each trip iterates over 19 stops.

In a practical setting, it is important that the implemented forecasting model is able to update within the time bounds of the link between consecutive stops. The timetable of line 11[1] shows durations between stops as low as one minute. Fortunately, all model forecasts will approximately fall below one minute overall. This implies that all of these models are suitable to deploy in a practical setting, provided that they are deployed on a machine with similar computing power[2].

	Method	Total Runtime	Runtime per Trip
Without Updating	SARIMA	44 minutes	23.27 seconds
	SARIMA, GARCH	46 minutes	24.38 seconds
	SARIMA, Exogenous	1 hour 23 minutes	43.81 seconds
	SARIMA, GARCH, Exogenous	1 hour 34 minutes	49.55 seconds
With Updating	SARIMA	3 hours 14 minutes	102.02 seconds
	SARIMA, GARCH	9 hours 48 minutes	309.32 seconds
	SARIMA, Exogenous	6 hours 27 minutes	204.95 seconds
	SARIMA, GARCH, Exogenous	19 hours 19 minutes	609.91 seconds

Table I.1: Runtimes of the different variations of the Seasonal ARIMA forecasting model. The approximate total runtimes have been rounded to a minute.

[1] Retrieved from `https://www.htm.nl/en/timetable/tram-11`

[2] The specific machine used in the evaluation is a `t3.small` Amazon EC2 instance. For specifications, see: `https://aws.amazon.com/ec2/instance-types/t3/`

Bibliography

[1] IEA, “Empowering Cities for a Net Zero Future,” IEA, Paris, Tech. Rep., 2021. [Online]. Available: https://www.iea.org/reports/empowering-cities-for-a-net-zero-future.

[2] D. L. Bleviss, “Transportation is critical to reducing greenhouse gas emissions in the United States,” *WIREs Energy and Environment*, vol. 10, no. 2, Mar. 2021, ISSN: 2041-8396. DOI: 10.1002/wene.390.

[3] E. Jenelius and M. Cebecauer, “Impacts of COVID-19 on public transport ridership in Sweden: Analysis of ticket validations, sales and passenger counts,” *Transportation Research Interdisciplinary Perspectives*, vol. 8, p. 100 242, 2020.

[4] L. Moreira-Matias and O. Cats, “Toward a Demand Estimation Model Based on Automated Vehicle Location,” *Transportation Research Record*, vol. 2544, no. 1, pp. 141–149, 2016. DOI: 10.3141/2544-16.

[5] W. Sun, J.-D. Schmöcker, and K. Fukuda, “Estimating the route-level passenger demand profile from bus dwell times,” *Transportation Research Part C: Emerging Technologies*, vol. 130, p. 103 273, 2021. DOI: 10.1016/j.trc.2021.103273.

[6] J. Zhang, D. Shen, L. Tu, F. Zhang, C. Xu, Y. Wang, C. Tian, X. Li, B. Huang, and Z. Li, “A Real-Time Passenger Flow Estimation and Prediction Method for Urban Bus Transit Systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3168–3178, 2017. DOI: 10.1109/TITS.2017.2686877.

[7] E. Jenelius, “Personalized predictive public transport crowding information with automated data sources,” *Transportation Research Part C: Emerging Technologies*, vol. 117, p. 102 647, 2020, ISSN: 0968-090X. DOI: 10.1016/j.trc.2020.102647.

[8] D. Luo, L. Bonnetain, O. Cats, and H. van Lint, “Constructing Spatiotemporal Load Profiles of Transit Vehicles with Multiple Data Sources,” *Transportation Research Record*, vol. 2672, no. 8, pp. 175–186, 2018. DOI: 10.1177/0361198118781166.

[9] E. Jenelius, “Data-driven metro train crowding prediction based on real-time load data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2254–2265, 2019.

[10] B. F. Nielsen, L. Frølich, O. A. Nielsen, and D. Filges, “Estimating passenger numbers in trains using existing weighing capabilities,” *Transportmetrica A: Transport Science*, vol. 10, no. 6, pp. 502–517, 2014. DOI: 10.1080/23249935.2013.795199.

[11] R. Kovacs, L. Nadai, and G. Horvath, “Concept validation of an automatic passenger counting system for trams,” in *2009 5th International Symposium on Applied Computational Intelligence and Informatics*, 2009, pp. 211–216. DOI: 10.1109/SACI.2009.5136243.

[12] Y. Ji, J. Zhao, Z. Zhang, and Y. Du, “Estimating bus loads and OD flows using location-stamped farebox and Wi-Fi signal data,” *Journal of Advanced Transportation*, vol. 2017, 2017.

[13] M. Nitti, F. Pinna, L. Pintor, V. Pilloni, and B. Barabino, “iabacus: A wi-fi-based automatic bus passenger counting system,” *Energies*, vol. 13, no. 6, p. 1446, 2020.

[14] M. Dunlap, Z. Li, K. Henrickson, and Y. Wang, “Estimation of origin and destination information from Bluetooth and Wi-Fi sensing for transit,” *Transportation Research Record*, vol. 2595, no. 1, pp. 11–17, 2016.

[15] Y. Maekawa, A. Uchiyama, H. Yamaguchi, and T. Higashino, “Car-level congestion and position estimation for railway trips using mobile phones,” in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014, pp. 939–950.

[16] L. Liu and R.-C. Chen, “A novel passenger flow prediction model using deep learning methods,” *Transportation Research Part C: Emerging Technologies*, vol. 84, pp. 74–91, 2017, ISSN: 0968-090X. DOI: 10.1016/j.trc.2017.08.001.

[17] F. Toqué, M. Khouadjia, E. Come, M. Trepanier, and L. Oukhellou, “Short & long term forecasting of multimodal transport passenger flows with machine learning methods,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 560–566. DOI: 10.1109/ITSC.2017.8317939.

[18] P. Wang, X. Chen, J. Chen, M. Hua, and Z. Pu, “A two-stage method for bus passenger load prediction using automatic passenger counting data,” *IET Intelligent Transport Systems*, vol. 15, no. 2, pp. 248–260, 2021.

[19] M. S. Ahmed and A. R. Cook, *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*, 722. 1979.

[20] C. Ding, J. Duan, Y. Zhang, X. Wu, and G. Yu, “Using an ARIMA-GARCH modeling approach to improve subway short-term ridership forecasting accounting for dynamic volatility,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1054–1064, 2017. DOI: 10.1109/tits.2017.2711046.

[21] M. Gong, X. Fei, Z. H. Wang, and Y. J. Qiu, “Sequential framework for short-term passenger flow prediction at bus stop,” *Transportation Research Record*, vol. 2417, no. 1, pp. 58–66, 2014.

[22] Y. Jia, P. He, S. Liu, and L. Cao, “A combined forecasting model for passenger flow based on GM and ARMA,” *International Journal of Hybrid Information Technology*, vol. 9, no. 2, pp. 215–226, 2016.

[23] W. Li, L. Sui, M. Zhou, and H. Dong, “Short-term passenger flow forecast for urban rail transit based on multi-source data,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–13, 2021.

[24] M. Milenković, L. Švadlenka, V. Melichar, N. Bojović, and Z. Avramović, “SARIMA modelling approach for railway passenger flow forecasting,” *Transport*, pp. 1–8, Mar. 2016, ISSN: 1648-4142. DOI: 10.3846/16484142.2016.1139623.

[25] E. Chen, Z. Ye, C. Wang, and M. Xu, “Subway Passenger Flow Prediction for Special Events Using Smart Card Data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1109–1120, Mar. 2020, ISSN: 1524-9050. DOI: 10.1109/TITS.2019.2902405.

[26] C. Chen, J. Hu, Q. Meng, and Y. Zhang, “Short-time traffic flow prediction with ARIMA-GARCH model,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Jun. 2011, pp. 607–612, ISBN: 978-1-4577-0890-9. DOI: 10.1109/IVS.2011.5940418.

[27] G. Bishop and G. Welch, “An introduction to the kalman filter,” *Proc of SIGGRAPH, Course*, vol. 8, no. 27599-23175, p. 41, 2001.

[28] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Oct. 1960, ISSN: 0021-9223. DOI: 10.1115/1.3662552.

[29] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[30] J. Guo, Z. Xie, Y. Qin, L. Jia, and Y. Wang, “Short-term abnormal passenger flow prediction based on the fusion of SVR and LSTM,” *IEEE Access*, vol. 7, pp. 42 946–42 955, 2019.

[31] K. Pasini, M. Khouadjia, A. Same, F. Ganansia, and L. Oukhellou, “LSTM encoder-predictor for short-term train load forecasting,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2019, pp. 535–551.

[32] S. Liyanage, R. Abduljabbar, H. Dia, and P.-W. Tsai, “AI-based neural network models for bus passenger demand forecasting using smart card data,” *Journal of Urban Management*, May 2022, ISSN: 22265856. DOI: 10.1016/j.jum.2022.05.002.

[33] Y. Liu, Z. Liu, and R. Jia, “DeepPF: A deep learning based architecture for metro passenger flow prediction,” *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 18–34, 2019, ISSN: 0968-090X. DOI: 10.1016/j.trc.2019.01.027.

[34] D. Yang, K. Chen, M. Yang, and X. Zhao, “Urban rail transit passenger flow forecast based on LSTM with enhanced long-term features,” *IET Intelligent Transport Systems*, vol. 13, no. 10, pp. 1475–1482, 2019.

[35] Y. C. Shiao, L. Liu, Q. Zhao, and R. C. Chen, “Predicting passenger flow using different influence factors for Taipei MRT system,” in *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*, IEEE, Nov. 2017, pp. 447–451, ISBN: 978-1-5386-2965-9. DOI: 10.1109/ICAwST.2017.8256497.

[36] L. Liu, R.-C. Chen, Q. Zhao, and S. Zhu, “Applying a multistage of input feature combination to random forest for improving MRT passenger flow prediction,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 11, pp. 4515–4532, Nov. 2019, ISSN: 1868-5137. DOI: 10.1007/s12652-018-1135-2.

[37] F. Gallo, F. Corman, and N. Sacco, “Real-time occupancy predictions of public transport vehicles,” in *22nd Swiss Transport Research Conference (STRC 2022)*, 2022.

[38] C. Ding, D. Wang, X. Ma, and H. Li, “Predicting Short-Term Subway Ridership and Prioritizing Its Influential Factors Using Gradient Boosting Decision Trees,” *Sustainability*, vol. 8, no. 11, p. 1100, Oct. 2016, ISSN: 2071-1050. DOI: 10.3390/su8111100.

[39] Y. Zhao, L. Ren, Z. Ma, and X. Jiang, “Novel Three-Stage Framework for Prioritizing and Selecting Feature Variables for Short-Term Metro Passenger Flow Prediction,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 8, pp. 192–205, Aug. 2020, ISSN: 0361-1981. DOI: 10.1177/0361198120926504.

[40] T. Chen and C. Guestrin, “XGBoost,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2016, pp. 785–794, ISBN: 9781450342322. DOI: 10.1145/2939672.2939785.

[41] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[42] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Annals of statistics*, pp. 1189–1232, 2001.

[43] Tin Kam Ho, “Random decision forests,” in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, IEEE Comput. Soc. Press, 1995, pp. 278–282, ISBN: 0-8186-7128-9. DOI: 10.1109/ICDAR.1995.598994.

[44] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification And Regression Trees*, 1st Edition. New York: Routledge, Oct. 1984, ISBN: 9781315139470. DOI: 10.1201/9781315139470.

[45] T. Hastie, R. Tibshirani, and J. Friedman, “Boosting and Additive Trees,” in 2009, pp. 337–387. DOI: 10.1007/978-0-387-84858-7.

[46] J. H. Friedman, “Stochastic gradient boosting,” *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, Feb. 2002, ISSN: 01679473. DOI: 10.1016/S0167-9473(01)00065-2.

[47] Z. C. Lipton, “The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery.,” *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[48] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, “A Survey of Methods for Explaining Black Box Models,” *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–42, Sep. 2019, ISSN: 0360-0300. DOI: 10.1145/3236009.

[49] S. M. Lundberg, G. G. Erion, and S.-I. Lee, “Consistent Individualized Feature Attribution for Tree Ensembles,” Feb. 2018.

[50] L. Auret and C. Aldrich, “Empirical comparison of tree ensemble variable importance measures,” *Chemometrics and Intelligent Laboratory Systems*, vol. 105, no. 2, pp. 157–170, Feb. 2011, ISSN: 01697439. DOI: 10.1016/j.chemolab.2010.12.004.

[51] S. Lin and H. Tian, “Short-Term Metro Passenger Flow Prediction Based on Random Forest and LSTM,” in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, Jun. 2020, pp. 2520–2526, ISBN: 978-1-7281-4390-3. DOI: 10.1109/ITNEC48623.2020.9084974.

[52] P. Wei, Z. Lu, and J. Song, “Variable importance analysis: A comprehensive review,” *Reliability Engineering & System Safety*, vol. 142, pp. 399–432, Oct. 2015, ISSN: 09518320. DOI: 10.1016/j.ress.2015.05.018.

[53] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, “Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation,” *Journal of Computational and Graphical Statistics*, vol. 24, no. 1, pp. 44–65, Jan. 2015, ISSN: 1061-8600. DOI: 10.1080/10618600.2014.907095.

[54] C. Molnar, *Interpretable Machine Learning*, 2nd ed. 2022. [Online]. Available: christophm.github.io/interpretable-ml-book/.

[55] M. T. Ribeiro, S. Singh, and C. Guestrin, “”Why Should I Trust You?”: Explaining the Predictions of Any Classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2016, pp. 1135–1144, ISBN: 9781450342322. DOI: 10.1145/2939672.2939778.

[56] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc., 2017.

[57] L. S. Shapley, *A value for n-person games, Contributions to the Theory of Games, 2, 307–317*, 1953.

[58] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[59] V. Kotu and B. Deshpande, *Data science: concepts and practice*, Second edition. Elsevier, 2019, pp. 395–445, ISBN: 978-0128147610.

[60] J. D. Hamilton, *Time series analysis*. Princeton university press, 1994.

[61] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. USA: Holden-Day, Inc., 1990, ISBN: 0816211043.

[62] P. J. Brockwell and R. A. Davis, *Time series: theory and methods*. Springer Science & Business Media, 2009, p. 253.

[63] S. Seabold and J. Perktold, “statsmodels: Econometric and statistical modeling with python,” in *9th Python in Science Conference*, 2010.

[64] G.-H. Han, J. Chung, and J. K. Yoo, “A study on prediction for attendances of Korean probaseball games using covariates,” *Journal of the Korean Data and Information Science Society*, vol. 25, no. 6, pp. 1481–1489, 2014.

[65] R. F. Engle, “Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation,” *Econometrica: Journal of the econometric society*, pp. 987–1007, 1982.

[66] T. Bollerslev, “Generalized autoregressive conditional heteroskedasticity,” *Journal of econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[67] R. Engle, “GARCH 101: The Use of ARCH/GARCH Models in Applied Econometrics,” *Journal of Economic Perspectives*, vol. 15, no. 4, pp. 157–168, Nov. 2001, ISSN: 0895-3309. DOI: 10.1257/jep.15.4.157.

[68] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948, ISSN: 00058580. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[69] Taylor G. Smith, *pmdarima: ARIMA estimators for Python*, 2017. [Online]. Available: http://www.alkaline-ml.com/pmdarima.

[70] W. Green, *Econometric analysis*, 5th edition. Upper Saddle River, NJ: Pearson Education, Inc, 2003.

[71] F. Toque, E. Come, M. K. El Mahrsi, and L. Oukhellou, “Forecasting dynamic public transport Origin-Destination matrices with long-Short term Memory recurrent neural networks,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, Nov. 2016, pp. 1071–1076, ISBN: 978-1-5090-1889-5. DOI: 10.1109/ITSC.2016.7795689.

[72] G. Van Rossum and F. L. Drake Jr, *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[73] W. Zhang, C. Zhang, and F. Tsung, “Transformer Based Spatial-Temporal Fusion Network for Metro Passenger Flow Forecasting,” in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, IEEE, Aug. 2021, ISBN: 978-1-6654-1873-7. DOI: 10.1109/CASE49439.2021.9551442.

[74] Y. Bai, Z. Sun, B. Zeng, J. Deng, and C. Li, “A multi-pattern deep fusion model for short-term bus passenger flow forecasting,” *Applied Soft Computing*, vol. 58, pp. 669–680, Sep. 2017, ISSN: 15684946. DOI: 10.1016/j.asoc.2017.05.011.

[75] L. Tang, Y. Zhao, J. Cabrera, J. Ma, and K. L. Tsui, “Forecasting Short-Term Passenger Flow: An Empirical Study on Shenzhen Metro,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3613–3622, Oct. 2019, ISSN: 1524-9050. DOI: 10.1109/TITS.2018.2879497.