Data Driven Mobility

Slik, Jesper Siem

2022

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

citation for published version (APA)
Slik, J. S. (2022). *Data Driven Mobility*. Ipskamp Drukkers.

Data Driven Mobility

Jesper Slik

Cover design by Menno Anker

Typeset by LaTeX

Printed by Ipskamp Printing

ISBN: 978-94-6421-720-9

VRIJE UNIVERSITEIT

Data Driven Mobility

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. J.J.G. Geurts,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Bètawetenschappen
op woensdag 18 mei 2022 om 11.45 uur
in een bijeenkomst van de universiteit,
De Boelelaan 1105

door

Jesper Siem Slik

geboren te Blaricum

promotoren:	prof.dr. S. Bhulai prof.dr. R.D. van der Mei
promotiecommissie:	prof.dr. G.M. Koole prof.dr.ir. R. Dekker prof.dr. S. Klous dr. E.R. Dugundji dr. A. Wünsch

"Remember kids, the only difference between screwing around and science is writing it down."

Adam Savage

Acknowledgements

This thesis would not have been possible without the support of others. I would like to express my gratitude towards those in this section.

Sandjai, your enthusiasm highly encouraged me to start pursuing my research and throughout the past four years to keep going in the right direction. And I have not regretted this decision. In our many weekly discussions we seemed to run out of time in all of them, topics ranging from research to events and basically anything non-work related. And I am glad more will follow in the future, as this dissertation is not the end of our collaboration. Perhaps I will have an influence on your life as well, as nowadays you are traveling to the university by bike.

Rob, your fast and sharp feedback helped in improving many parts of this dissertation. During the time we could visit the office, the small talk in the hallways was always enjoyable. Besides, I would like to express my gratitude towards the dissertation committee for their time and efforts in reading and evaluating my dissertation.

Ralf, your opportunism is the reason this thesis was executed in collaboration with Pon. You can sense where opportunities lie, and most importantly, find a way somehow to make them work. I am glad most of our conversations were aimed towards those opportunities and that we made most of them work. Despite having boring conversations on dashboarding software lately, I am looking forward to initiating more algorithms within Pon.

Kevin, I think your enthusiasm exceeds those of all others mentioned in this section. However, I much appreciate this was later accompanied by support on growing myself on the non-technical skills. Finding the questions behind the questions. Besides, I much enjoyed your energy during drinks and even some sport events.

Furthermore, I would like to thank many Pon colleagues. Firstly, Bas and Ton for providing the financial trust required for setting up the first PhD collaboration within Pon. Besides, I would like to thank all team members of the Datalab for the great working environment, implementing results of this research, contributions to brainstorms, and the slight competition during team events. It has been a pleasure to watch the team grow to its current state.

The A&O group of the VU has always been a pleasure to be a part of, despite 1.5 years of corona lockdowns. Special thanks to Ger for initiating the various hiking trips and events, Alessandro for the alpine lessons, and Jaap for the bouldering practices. Besides, many thanks for the many conversations with roomie Siqiao, Anni, Robin van Ruitenbeek, Guus, and Rik.

Furthermore, I want to thank all members of the Greyhound athletics group for allowing me to stop thinking sometimes. Special thanks to the coach Erik for teaching me to run properly and Miljo for pushing me during the intervals.

Additionally, I have various friends to thank for keeping my mental well-being. Noam and Wilte for the good dinners and distractions through side projects. Furthermore, Roy and Nino for allowing me too keep my mountainbiking skills to a minimum level. Besides, Maarten and Ab for the sarcastic conversations with sometimes questionable humour. And finally Youri. Despite you are no longer with us, I look back on our many nights of gaming with gratitude.

Last but not least, I want to thank my family. Ma^2 and pa^2 for the free food during the weekends, Rozanne for the plants in my apartment, and Ivar for the high quality cycling socks. All kidding aside, you were a great support.

Contents

1 Introduction

Each minute in 2020, over 250,000 online meetings were held, more than 500 hours of video were uploaded, and USD 1M was spent online [67]. At the end of the year, approximately 64 ZB of data were created [135]. While you were reading this introduction so far, more than 21 PB of data have been generated. This is roughly equal to the storage of 21,000 high-end laptops.[1] Despite these being estimates, a large amount of data is being generated. Data is becoming a major part of our lives, and a continued growth is expected. But why do we need such a vast amount of data? And how can we take advantage of it?

Data is the oil of the 21st century, according to various experts around the world [9, 140, 30]. It promises to support any organization in making better decisions. Thus, its applications are not bound to a specific sector. Examples are: cancer detection, supply chain optimization, vehicular automation, churn prediction, or speech recognition. These seem to have little in common, however, the underlying technologies are similar. Combining all applications, it is estimated the big data and business analytics market will be valued at USD 274 billion by 2022 [134].

Data is a resource, like oil, which does little on its own. It needs to be converted to information, knowledge, or wisdom to deliver value. Some data is easy to process. For example, the data generated by a motion sensor can directly be used to turn on a bathroom light. Other data is difficult to interpret. For example, should an autonomous vehicle brake when it observes a nearby pedestrian? When the pedestrian is expected to cross paths with the vehicle, it should. But no sensor exists which directly predicts human behavior. Thus, additional data processing is required. Through recent developments in the fields of mathematics and computer science, nowadays we can interpret more data. More in terms of all V's of big data: volume, variety, velocity, and veracity.

To appropriately convert data, different methods suit different scenarios. A common approach is to classify methods on their purpose. A distinction can be made between descriptive, predictive, and prescriptive analytics. *Descriptive* analytics attempts to describe what happened in the past. For example, computing financial metrics to describe how well an organization performed. *Predictive* analytics attempts to predict what is going to happen. For example, predicting the sales volume of any product for the upcoming year. *Prescriptive* analytics attempts to prescribe what action to execute. For example, prescribing that a company should produce more bicycles, as this market segment is expected to grow.

Despite its growth and recent technological developments, data remains a challenge for decision makers. Unlike oil, data is practically infinite, reusable, and becoming increasingly available. Additionally, substantial investment is often required before the value of data is certain or even recognized. The main challenge lies in identifying decisions and designing methodology for direct support. Auxiliary challenges

[1] A laptop having 1TB of storage; with an average reading speed of 238 words per minute [17].

include data integration, analytical skills, security and privacy, infrastructure, and synchronization [123].

In this dissertation, we propose various methods to interpret diverse data. We contribute across the analytics spectrum with descriptive, predictive, and prescriptive analyses. We implement part of these analyses through industry partnerships and achieve results in practice. As these companies primarily operate in the mobility industry, most chapters in this dissertation focus on mobility. In the remainder of this chapter, we introduce each of these analyses and highlight our main contributions. Each chapter can be read on its own and independently of one another.

1.1 Descriptive Analyses

The first category within the analytics framework is descriptive analytics. As stated in the previous section, it concerns describing what happened. Typically, this consists of registering, storing, and presenting data as accurately as possible. Most of these tasks are executed by traditional business intelligence departments. However, sometimes it is challenging to describe what happened and the raw data needs to be processed. This could be caused by data quality issues such as missing data or human mistakes. Additionally, the raw data format might not be insightful or answer any questions.

In this section, we introduce three descriptive analyses. The first concerns detecting outliers in univariate time series, the second understanding human mobility choices, and the third measuring the global effect of Covid-19 on human mobility.

Chapter 2 - Detection of Additive Outliers in Univariate Time Series

Describing what happened is straightforward if the collected data perfectly describes so. However, this is often not what happens in practice. The inspiration for this research is keeping track of inventory levels across stores scattered throughout the Netherlands. Each store reports daily inventory levels. This generates a one-dimensional series ordered in time. However, through system failures and communication issues, data might not be reported or sent in duplicate. This generates so-called *additive outliers*. We aim to detect these and improve data quality by resolving issues they create.

In this research, we perceive an additive outlier as a surprisingly large or small value occurring for a single observation in a time series. The detection of these outliers is an important issue because their presence may have serious negative effects on the analysis in many different ways. Existing methods to detect such outliers are often inadequate due to poor accuracy, high complexity, and long runtimes. We provide a novel approach to detect additive outliers that overcomes these drawbacks. We validate our approach by comparing against existing techniques and benchmark performance. Experimental results on benchmark datasets show that our proposed technique outperforms existing methods on several measures.

Chapter 2 is based on Slik and Bhulai (2021): *Detection of Additive Outliers in Univariate Time Series* [127]. Currently in review.

Chapter 3 – Understanding Human Mobility for Data-Driven Policy Making

An advantage of collaboration with the industry is having access to unique datasets. For this research, we were granted access to a rich dataset containing mobility transactions. These contain daily choices made by individuals with respect to their travel behavior by car or public transport. Combining this with publicly available data related to congestion, we give a unique description of human mobility behavior.

In this research, we aim to identify the patterns of behavior which underlie human mobility. More specifically, we compare commuters who drive in a car with those who use the train in the same geographic region of the Netherlands. We try to understand the mode choices of the commuters based on three factors: the cost of the transport mode, the CO_2 emissions, and the travel time. The analysis has been based on data consisting of travel transactions in the Netherlands during 2018 containing over half a million records. We show how this raw data can be transformed into relevant insights on the three factors. A large difference is observed in terms of CO_2 emissions and cost, a minor difference in speed. Besides, the computation of congestion shows intuitive results. These results can be used to stimulate behavioral change proactively and to improve trip planners.

Chapter 3 is based on Slik and Bhulai (2021): *Understanding Human Mobility for Data-Driven Policy Making* [131]. Currently in review.

Chapter 4 – On the Relation between Covid-19, Mobility, and the Stock Market

This dissertation was written in turbulent times, as a result of the worldwide Covid-19 outbreak. This outbreak has consequences on many aspects of human behavior, including mobility usage. Through this chapter, we describe the effects of the Covid-19 outbreak on mobility usage and the impact of Covid-19 measures undertaken. We measure mobility usage globally in terms of vessels, flights, and vehicle activity combined with train and bicycle online search behavior.

The Covid-19 pandemic has brought forth a major landscape shock in the mobility sector. Due to its recency, researchers have just started studying and understanding the implications of this crisis on mobility. We contribute by combining mobility data from various sources to bring a novel angle to understanding mobility patterns during Covid-19. The goal is to expose relations between mobility and Covid-19 variables and understand them by using our data. Amongst other findings, we observe the usage of all mobility types, except bicycle, has declined since March 2020. Besides, we observe significant correlations between the investigated variables. This is crucial information for governments to understand and address the underlying root causes of the impact.

In this research, we argue the Covid-19 pandemic has brought forth a major landscape shock in the mobility sector. Due to its recency, researchers have just started studying and understanding the implications of this crisis on mobility. We contribute by combining mobility data from various sources to bring a novel angle to understanding mobility patterns during Covid-19. The goal is to expose relations between mobility and Covid-19 variables and understand them by using our data. Amongst other findings, we observe the usage of all mobility types, except bicycle, has declined since March 2020. Besides, we observe significant correlations between the investigated variables. This is crucial information for governments to understand and address the underlying root causes of the impact.

Chapter 4 is based on van Ruitenbeek, Slik, and Bhulai (2021): *On the Relation between Covid-19, Mobility, and the Stock Market* [109]. Published in PLOS ONE.

1.2 Predictive Analyses

The second category within the analytics framework is predictive analytics. These analyses build on the descriptive analyses and go one step further. After describing what happened, they aim to predict what is going to happen. This requires different methodology and a slightly different point of view. A major challenge in any predictive analysis is to balance historic performance (train), and future performance (test). Often, it is relatively easy to gain a high train performance through overfitting. However, a proper predictive analysis balances train and test performance and generates reliable results.

In this section, we introduce three predictive analyses. The first concerns predicting the travel behavior of individuals. The second aims to predict the effects of changing physical store locations on sales volumes. The third analysis aims to forecast future sales.

Chapter 5 – Predicting Travel Behavior by Analyzing Mobility Transactions

After describing human mobility behavior in Chapter 2, we extend this research toward predicting this behavior in the future. All mobility choices made by the many individuals in the dataset are analyzed on an aggregated level to take privacy into account. The resulting model can be used to help human decision making, by proposing the right mobility types for any requested travel plan. Currently, mobility types include public transport and car, but this can be extended towards shared mobility services.

In this research, we argue that urban planning can benefit tremendously from a better understanding of *where*, *when*, *why*, and *how* people travel. Through advances in technology, detailed data on the travel behavior of individuals has become available. This data can be leveraged to understand why one prefers one mode of transportation over another. We analyze a unique dataset through which we can address this question. We show that the travel behavior in our dataset is highly predictable, with an accuracy of 97%. The main predictors are reachability features, more so than specific travel times. Moreover, the travel type (commute or personal) has a considerable influence on travel mode choice.

Chapter 5 is based on Slik and Bhulai (2020): *Predicting Travel Behavior by Analyzing Mobility Transactions* [129]. Published in the Journal of Traffic and Transportation Management.

Chapter 6 – Accessibility Analysis for Private Car and Public Transport: Comparable Measures for Data-Driven Policymaking

Predicting human behavior can improve a wide range of business decisions. Extending our mobility-related analysis, we aim to predict the effects of physical store placement on sales volume. A balance must be struck between placing stores and reaching individuals swiftly. Placing too many stores results in a large operational cost, however, placing too few stores results in lost sales. We achieve a balance by thoroughly analyzing travel duration, predicting the willingness to travel, and predicting network effects such as cannibalism.

In this research, we argue that the disparity between the accessibility of areas through different travel modes is essential for the choice of the mode of transport. Calculation of the travel times by different travel modes is, therefore, very important. Many urban design decisions on infrastructure depend on these calculations. Developments in open data policies among urban data producers make this analysis more tractable. In this chapter, we apply a data-driven approach to travel time estimation based on realized past travel times. We compare commuters who drive in a car with those who use the train in the same geographic region of the Netherlands. First, we propose a method to quantify the accessibility of areas for these different modalities. Second, we show how these metrics can be used to determine optimal locations based on the willingness to travel. The results can be integrated into planning software to making data-driving decisions for policymaking.

Chapter 6 is based on Slik and Bhulai (2021): *Accessibility Analysis for Private Car and Public Transport: Comparable Measures for Data-Driven Policymaking* [124]. Currently in review.

Chapter 7 – Overcoming the Self-Fulfilling Prophecy in Time Series Forecasting

A common use case for predictive analysis is sales forecasting. Any organization dependent on product sales would benefit tremendously from knowing what their customers require in the coming weeks, months, or years. Various decisions can be improved by knowing what is going to happen. However, the future is uncertain. And this uncertainty might vary across sectors or product groups. Sales forecasting aims to predict future sales as accurately as possible through finding and extrapolating patterns in historic sales data. We contribute to this field by finding robust seasonal patterns through applying hierarchical clustering.

In this research, we observe two current challenges in time series forecasting are the *self-fulfilling prophecy* and finding *robust seasonal patterns*. We argue that both can be overcome through combining similar time series. We propose a new methodology to extract robust seasonal patterns from low-level sales data through applying hierarchical clustering. We validate our approach using a simulation experiment and a real-life dataset containing over €2B of bicycle sales. Our simulation results show a 45% decrease in forecasting error and they quantify the effects of the self-fulfilling prophecy on forecasting error. Our results on real-life data show a 15% performance gain on the benchmark when applying clustering. Additionally, we show insights in the effects of applying smoothing and forecasting sell-in vs sell-out data.

Chapter 7 is based on Slik and Bhulai (2021): *Overcoming the Self-Fulfilling Prophecy in Time Series Forecasting* [128]. Currently in review.

1.3 Prescriptive Analyses

The third category within the analytics framework is prescriptive analytics. These analyses build on the predictive analyses and again go one step further. After predicting what will happen, they aim to prescribe what to do. Thus, a prescriptive analysis partly contains a descriptive and a predictive analysis. Converting the predictions to actions also requires a slightly different point of view. Interesting challenges arise, such as the exploration-exploitation trade-off. It seems tempting to fully exploit current knowledge and only execute the best action. However, it might be better to explore other actions and learn from their consequences. In the long run,

a well-balanced approach will find the best possible action and adapt to a changing environment.

In this section, we introduce two prescriptive analyses. The first concerns prescribing which email to send to which person at what time. The second concerns prescribing to a robot which actions to take in order to perform a task.

Chapter 8 – Approximate Dynamic Programming for Optimal Direct Marketing

As illustrated in the first paragraph of this introduction, a large part of communication nowadays is digital. Especially for corporations, email has grown to become an important channel. Interestingly, any email can be tracked by using so-called tracker pixels. By doing so, the company can measure whether an email was opened, interacted with, or whether it resulted in an online purchase. This data is highly suitable for analysis. We aim to improve email marketing effectiveness through prescribing which email to send next on an individual basis. Based on the historic behavior of each user, we predict its interest in various email types and subsequently prescribe which email to send next.

In this research, we argue that email marketing is a widely used business tool that is in danger of being overrun by unwanted commercial email. Therefore, direct marketing via email is usually seen as notoriously difficult. One needs to decide which email to send at what time to which customer in order to maximize the email interaction rate. Two main perspectives can be distinguished: scoring the relevancy of each email and sending the most relevant, or seeing the problem as a sequential decision problem and sending emails according to a multi-stage strategy. In this chapter, we adopt the second approach and model the problem as a Markov decision problem (MDP). The advantage of this approach is that it can balance short- and long-term rewards and allows for complex strategies. We illustrate how the problem can be modeled such that the MDP remains tractable for large datasets. Furthermore, we numerically demonstrate by using real data that the optimal strategy has a high interaction probability, which is much higher than a greedy strategy or a random strategy. Therefore, the model leads to better relevancy to the customer and thereby generates more revenue for the company.

Chapter 8 is based on Slik and Bhulai (2020): *Approximate Dynamic Programming for Optimal Direct Marketing* [125]. Published in the International Journal on Advances in Internet Technology.

Chapter 9 – Benefits of Social Learning in Physical Robots

The final chapter in this dissertation is unique, as the algorithm's actions are directly executed in real life. It concerns controlling a robot in a protected environment. The robot has to execute a task, however, it needs to learn itself how to do so. It does so by 'trying' different actions and learning which actions are useful in which situation. This behavior is learned and stored in a neural network, which is evolved over generations through an evolutionary algorithm. We combine the experience of a single robot with others, such that they can learn socially and in parallel.

In this research, we focus on robot-to-robot learning. This is a specific case of social learning in robotics that enables the ability to transfer robot controllers directly from one robot to another. Previous studies showed that the exchange of controller information can increase learning speed and performance. However, most of these studies have been performed in simulation, where robots are identical. Therefore, the results do not necessarily transfer to a real environment, where each robot is unique

by definition due to the random differences in hardware. In this chapter, we investigate the effect of exchanging controller information, on top of individual learning, in a group of Thymio II robots for two tasks: obstacle avoidance and foraging. The controllers of the robots are neural networks that evolve using a modified version of the state-of-the-art NEAT algorithm, called *cNEAT*, which allows the conversion of innovations numbers from other robots. This research shows that robot-to-robot learning seems to at least parallelize the search, reducing wall clock time. Additionally, controllers are less complex, resulting in a smaller search space.

Chapter 9 is based on Heinerman et al. (2018): *Benefits of Social Learning in Physical Robots* [54]. Published in the Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018.

Scientific publications not contained in this dissertation:

- Slik and Bhulai (2019): *Data-Driven Direct Marketing via Approximate Dynamic Programming* [126].
- Slik and Bhulai (2020). *Transaction-Driven Mobility Analysis for Travel Mode Choices* [130].

2 Detection of Additive Outliers in Univariate Time Series

2.1 Summary

An additive outlier appears as a surprisingly large or small value occurring for a single observation in a time series. The detection of these outliers is an important issue because their presence may have serious negative effects on the analysis in many different ways. Existing methods to detect such outliers are inadequate due to poor accuracy, high complexity, and long runtimes. In this chapter, we provide a novel approach to detect additive outliers that overcomes the mentioned drawbacks. We validate our approach by comparing against existing techniques and benchmark performance. Experimental results on benchmark datasets show that our proposed technique outperforms existing methods on several measures.

2.2 Introduction

Outliers have been an actively studied research topic in the past years [24, 99, 84]. Through the exponential growth in data collected worldwide, data can represent a wide range of real-life events [138]. Outliers might play a significant role in interpreting these data. Use cases include detection of fraud, intrusion, or faults, medical informatics, monitoring traffic, and many more. Typically, outliers can be thought of as *observations that do not follow the expected behavior* [15]. Various methodologies have been proposed on this basis. However, this definition poses challenges. How to define the expected behavior? And most importantly, who is to account for the deviation? Is it the data or the expectation?

In this chapter, we propose a novel detection method of outliers based on the difference between subsequent values in the series. The method defines an outlier as *a large change followed by a large opposite change*. This scopes the methodology to so-called additive outliers, or spikes. We show that the proposed methodology has the advantages of being intuitive, model-free, accurate, robust, and computationally inexpensive. Disadvantages of the method are its applicability to additive outliers only, and its dependence on two parameters that need to be set appropriately. We compare our methodology amongst three state-of-the-art methods on nine real-life use cases using hand-labeled definitions classified by five human experts.

Outliers in time series have typically been classified into five categories: Additive Outlier (AO), Innovative Outlier (IO), Level Shift (LS), Transitory Change (TC), and sometimes Seasonal Level Shift (SLS). As described in [71], an AO represents an isolated spike, an LS a step function, a TC a spike that takes a few periods to disappear, an IO effects that appear depending on the fitted ARIMA model, and an SLS

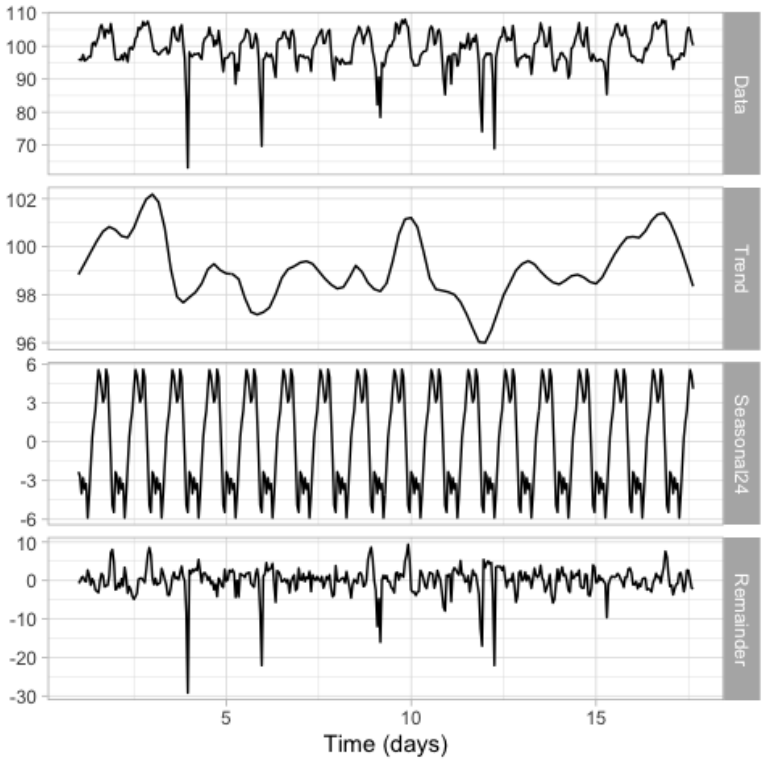


FIGURE 2.1: A time series decomposed in three parts: trend, seasonal, and the remainder. The raw data represents hourly averaged traffic speed on a highway segment in the Netherlands and spans approximately seventeen days.

a seasonally occurring level shift. An AO is typically related to missing or deleted observations [84].

A main approach is to fit a model to the time series, which then is used as the expectation. Data can be compared against this expectation, and (probabilistic) approaches can be used to determine whether the difference is significant. Typically, an ARMA, ARIMA, or SARIMA model is used to explain the data. The drawbacks of this approach are that computing such a model might be computationally expensive and that model itself might be influenced by outliers.

Various window-based approaches have been proposed in recent years. These methods do not aim to fit a model to the complete dataset. However, they use a sliding window to generate an expectation. Methods include nearest neighbor methods, such as K-Nearest Neighbors (KNN) or variations like KNN-CAD [42], and variations on Moving Averages (MA), such as EWMA, PEWMA, SD-EWMA, or TSSD-EWMA [100]. Drawbacks of these approaches are the imposed parameters, the influence of outliers on the expectation, and nearest neighbor methods do not take into account the ordering of the series. We develop a method that does not suffer from these drawbacks.

The structure of this paper is as follows. First, we define our methodology in Section 2.3 and highlight our design choices. Afterward, in Section 2.4, we illustrate which methodologies we compare against what in the experimental setup. Next, we describe the datasets used to evaluate our methodology in Section 2.5. Subsequently,

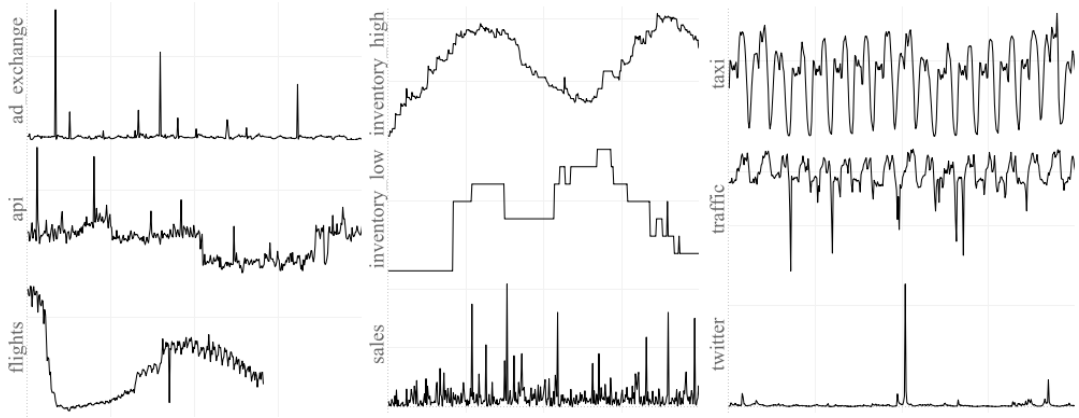


FIGURE 2.2: Overview of the benchmark datasets described in Section 2.5.

we describe our results in Section 2.6. Finally, Section 2.7 discusses our results, remarks, and future work.

2.3 Methodology

We define an additive outlier as *a large change followed by a large opposite change*. In our view, this is an intuitive definition which we will quantify in this section. Critically, the definition consists of two parts, each in the opposite direction. By explicitly using the time component of the series in this manner, the definition will produce robust scores. We can construct a score for each data point in the series by following the equations below. The intuition behind and explanation of each step is mentioned thereafter.

Consider time series $x_1, \ldots, x_T \in \mathbb{R}$ of length T. First, we decompose the series into trend, seasonal, and remainder components and remove the seasonal component. This leaves us with the series $y_1, \ldots, y_T \in \mathbb{R}$. We argue that removing the seasonal component can aid in detecting outliers, as the seasonal effect can explain part of the variability of the data. Much research has been devoted to decomposing time series, and robust approaches are available in open-source software such as in [64]. We follow the equations below to compute our outlier score. In these equations, the median absolute value of series y is noted as $|\tilde{y}|$ and $q_x(\alpha)$ represents the α-quantile of series x.

$$c_{t,\beta} = \beta \frac{y_t - y_{t-1}}{\max\{|y_t|, |y_{t-1}|\}} + (1-\beta)\frac{y_t - y_{t-1}}{|\tilde{y}|} \tag{2.1}$$

$$\check{c}_\alpha = q_{c_{t,\beta}}(\alpha) \tag{2.2}$$

$$c_{t,\alpha,\beta} = \begin{cases} c_{t,\beta} & \text{if } c_{t,\beta} < \check{c}_{0.25} - \alpha(\check{c}_{0.75} - \check{c}_{0.25}) \\ c_{t,\beta} & \text{if } c_{t,\beta} > \check{c}_{0.75} + \alpha(\check{c}_{0.75} - \check{c}_{0.25}) \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

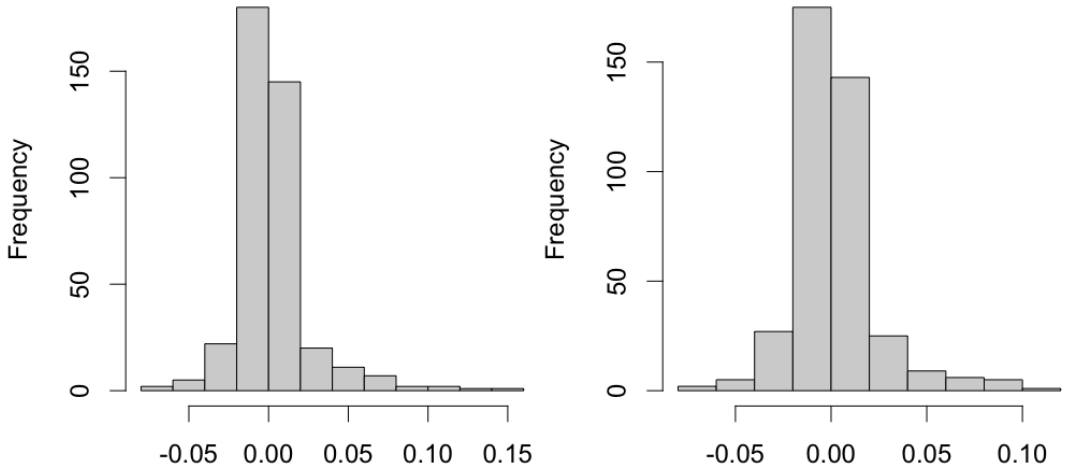


FIGURE 2.3: Distributions of local (left) and global (right) change for use case *inventory high*.

$$s_{t,\alpha,\beta} = \max\{0, c_{t,\alpha,\beta} \cdot -c_{t+1,\alpha,\beta}\} \tag{2.4}$$

Figure 2.1 visualizes the decomposition of one of the time series considered later in this chapter, named *traffic*. If the decomposition is proper, the remainder of the fully decomposed time series is stationary. Therefore, we check if the remainder is stationary through statistical tests before subtracting the seasonal pattern.

Equation 2.1 describes the change. In its basis, this is the difference between y_t and y_{t-1}. However, this absolute change is not adequate; more insightful is a relative change. Therefore, we compute and balance two relative changes: one relative to its previous value (*local*) and one relative to all known values (*global*). These local and global are balanced through parameter β; $0 \leq \beta \leq 1$. As described in [142], an adequate metric to measure the relative difference between x and y is a function $C : R^2 \rightarrow R$ having the following properties:

1. $C(x, y) = 0 \iff x = y$
2. $C(x, y) > 0 \iff x < y$
3. $C(x, y) < 0 \iff x > y$
4. C is a continuous and increasing function of x when y is fixed
5. $C(x, y) = C(ax, ay), \forall a : a > 0$
6. $C(x, y) = -C(y, x)$

The local measure for relative change satisfies all properties, the global all except the fifth. To ensure the measures are defined for positive and negative values of x and y, we divide by absolute values. For the global change, we compare against the median absolute value ($|\tilde{y}|$) to ensure the scale of the relative and absolute measures are comparable. Generally, this balances both changes well, as Figure 2.3 displays for a single use case. Also, we prefer the median over the mean as it is more robust to outliers.

Equation 2.2 describes the α-quantile of $c_{t,\beta}$.

Equation 2.3 describes a large change. Basically, we set the score to 0 when it lies within factor α of its interquartile range (IQR). The IQR is a commonly used measure for classifying outliers. It is a robust measure which we prefer above comparing against the standard deviation, as the impact of outliers on the standard deviation can be large.

Equation 2.4 describes a large change followed by a large opposite change. This is the final score we use in the classification of outliers. Intuitively, it is simply a multiplication of the previous equation at time t, time $t+1$, and the number -1. If they are in the opposite direction, a positive number arises; if not, a negative number or 0. We are not interested in values smaller than 0. Therefore, we take the maximum of the score and 0. The resulting score equals 0 if both changes are in a similar direction or if either one of them is within α of its IQR. The score is positive if both changes are outside α of its IQR and in the opposite direction. A higher score implies a larger outlier.

The code implementation in R is as follows:

```
score = function(x, alpha, beta, seasonal_period){
  x = remove_seasonal(x, seasonal_period)
  x_bwd = shift(x, n=1, fill=0)

  change_local = (x - x_bwd) / pmax(abs(x), abs(x_bwd))
  change_global = (x - x_bwd) / median(abs(x))
  change = beta * change_local + (1 - beta) * change_global

  q_25 = quantile(change, .25)[[1]]
  q_75 = quantile(change, .75)[[1]]
  change[which(q_25 - alpha * (q_75 - q_25) < change \
        & change < q_75 + alpha * (q_75 - q_25))] = 0
  change[c(1, length(change))] = 0  # boundaries

  score = pmax(0, -1*change*shift(change, n=-1, fill=0))
  return(score)
}
```

2.4 Experimental Setup

To evaluate our proposed methodology, we define the following experimental setup. First, we select nine time series originating from real-life use cases. Next, we implement our method in R and compare it against three well-known benchmark methods. Further, we define a ground truth by relying on the opinion of five human experts. Following, we classify all series in an online manner. Finally, we measure the precision, recall, and F1-score and compare the performance amongst all models accordingly.

The time series selected are described in detail in Section 2.5. As we require experts to hand label outliers, we do process the raw data to the aggregation level presented in Table 2.1. Additionally, we limit the length of each series by filtering on the latest known 400 data points.

The benchmarks used are Chen's approach as proposed in [25], PEWMA as proposed in [21], and KNN-CAD as proposed in [18]. These methods are implemented

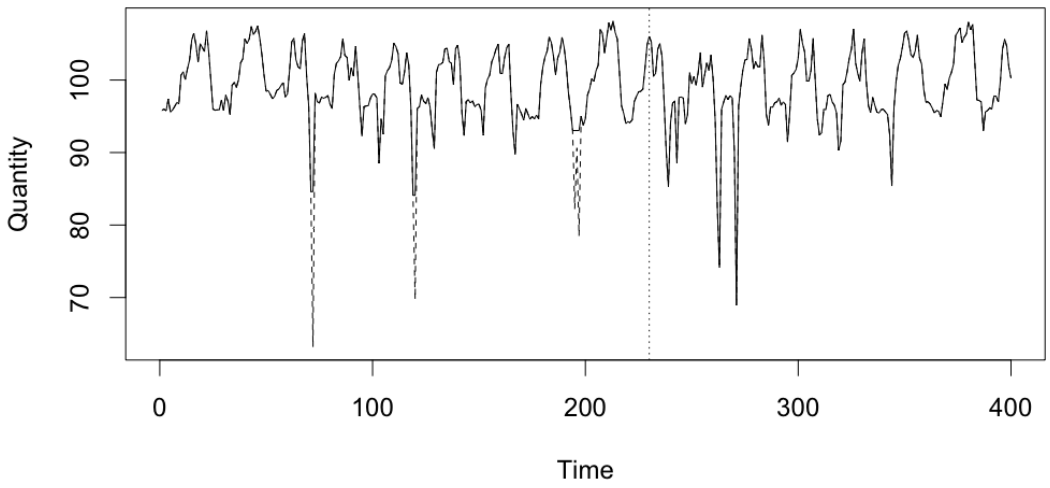


FIGURE 2.4: Online evaluation of use case *traffic* at timestamp 230 (vertical line). Outliers classified so far (dotted lines) are imputed.

in the R libraries *tsoutliers* and *otsad*. Both are considered well known, also represented through the fact that only these two packages are mentioned in the overview presented in [63]. As all methods depend on their parameter settings, we run many different parameters settings through an elaborate grid search and report on two versions: (1) default, having the highest average score over all time series, and (2) optimized, having the highest score optimized per time series. The highest score is with respect to our evaluation criterion. We use the optimized implementation of the PEWMA algorithm, through the *OipPewma* function.

The ground truth is required, as each method might have a different definition of an outlier, which might result in different outlier classifications. We establish a committee of five human experts who each hand label all use cases to additive outliers. Each expert works at Vrije Universiteit Amsterdam and has experience in the field of time series analysis. Their task was simply to label the outliers in all series, without them having knowledge of the methodology presented in this chapter. A majority vote is applied to create a final classification.

Most methods are evaluated in an online manner, as this most closely represents their performance when implemented in real life. We focus on applications in which timely classifying outliers is of the essence. An online evaluation implies splitting the data in a train- and test set based upon time. The test set consists of the latest known observations. The PEWMA and KNN-CAD methods have an online implementation built-in. For our method, we create a test set of size 10. We impute classified outliers in the train set with the previously known value, as visualized in Figure 2.4. Due to runtime considerations, we do not recreate an online version of Chen's approach.

The performance is measured through computing the precision, recall, and F1-score. These metrics are commonly used in binary classification problems. Each data point in the time series will be classified as either outlier or not. We can count the True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) rates. Then, the metrics are defined as follows:

TABLE 2.1: Characteristics of all data sets. KPSS and ADF indicate their respective p-values on the raw data, KPSS* and ADF* on the remainder of the decomposed time series.

.

use case	interval	n	min	mean	median	max	KPSS	KPSS*	ADF	ADF*
api	hour	11160	0	$2.65e+3$	$2.57e+3$	78853	< 0.01	> 0.1	0.582	< 0.01
taxi	hour	10320	8	$1.51e+4$	$1.68e+4$	39197	> 0.1	> 0.1	< 0.01	< 0.01
sales	day	1514	0	$7.25e+1$	$2.80e+1$	3036	> 0.1	> 0.1	< 0.01	< 0.01
traffic	hour	21517	7.8	$9.86e+1$	$9.85e+1$	188	> 0.1	> 0.1	< 0.01	< 0.01
flights	day	282	36	$2.41e+2$	$2.41e+2$	654	0.014	> 0.1	0.056	< 0.01
twitter	hour	15902	0	$8.56e+1$	$4.70e+1$	13479	> 0.1	> 0.1	< 0.01	< 0.01
ad exchange	hour	1643	0.024	$8.64e-2$	$7.28e-2$	3.1269	> 0.1	> 0.1	< 0.01	< 0.01
inventory low	day	653	1	$3.59e+0$	$4.00e+0$	8	< 0.01	> 0.1	0.908	< 0.01
inventory high	day	560	36	$9.59e+1$	$9.90e+1$	154	< 0.01	> 0.1	0.588	< 0.01

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.5}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.6}$$

$$\text{F1} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \tag{2.7}$$

In Section 2.3, we advise running our approach on the raw series minus the seasonal component. We implement the decomposition of the series through the *mstl* function of [64]. To validate the decomposition, we use two statistical tests: the KPSS and the ADF test. The KPSS test, proposed in [76], tests the null hypothesis that the time series is stationary around a deterministic trend. The ADF test has an alternative hypothesis that the time series is stationary.

Last, we execute an experiment to investigate the running time of each method as a function of the time series length. To do so, we generate a seasonal univariate time series of length N using the methods *stsm.model* and *datagen.stsm* as described in [77]. We generate a time series of length $1e + 07$ and classify outliers for each method using an increasing length of the series. We stop if the runtime exceeds the threshold of 60 seconds and repeat the experiment ten times. We cannot execute the experiment on a longer series due to the limited amount of memory available to our computer.

2.5 Data

We collect nine time series from various sources to evaluate the performance of our methodology. These sources span a wide range of real-life applications and are all generated by real-life events. Figure 2.2 gives an overview of the shape of each series.

As the origin of each series is unique, each series has different characteristics. Table 2.1 describes these in an aggregated manner. It displays the number of observations, interval level, general statistics, and the p-values of the KPSS and ADF stationarity tests after decomposing the series. The remainder of all datasets seem stationary, as for all KPSS tests, we do not reject stationarity, and for all ADF tests, we do reject non-stationarity.

TABLE 2.2: Performance (F1-score) by method and use case. Two parameter settings are displayed: one optimized for having the highest average over all cases, the other for having the highest score per case (indicated by *).

use case	tsoutlier	tsoutlier*	KNN-CAD	KNN-CAD*	PEWMA	PEWMA*	custom	custom*
api	0.89	0.91	0.50	0.50	0.60	1.00	0.89	0.89
taxi	0.00	0.00	0.00	0.14	1.00	1.00	1.00	1.00
sales	0.73	1.00	0.40	0.57	0.67	0.88	0.86	0.96
traffic	0.73	0.80	0.14	0.24	0.00	0.63	0.80	0.92
flights	0.73	0.80	0.18	0.20	0.50	0.83	0.29	0.83
twitter	0.89	0.89	0.29	0.43	0.63	0.91	0.36	0.75
ad exchange	1.00	1.00	0.44	0.44	0.87	0.96	0.77	1.00
inventory low	0.00	0.00	0.00	0.17	0.21	0.24	1.00	1.00
inventory high	1.00	1.00	0.29	0.29	0.21	0.50	0.80	1.00
average	0.69	0.71	0.25	0.33	0.52	0.77	0.75	0.93

The first use case, *ad exchange*, consists of online advertisement clicking rates, measured by cost-per-click (CPC). The data originates from the Numunta Anomaly Benchmark (NAB), as presented in [78]. These rates are measured hourly and are expressed by float values. At specific times, these rates might be unexpectedly high for various reasons. The data contains a seasonal pattern on a daily level.

The *api* use case consists of data describing the average duration of API calls of a critical system in the hospitality industry. It was provided to us on an aggregated level through the company maintaining these systems. Monitoring these API calls is critical in detecting the downtime of their services.

The *flights* use case describes daily flights departing from Schiphol airport in Amsterdam. The time window is February 2020 until August 2020, spanning the first months of the Covid-19 outbreak in the Netherlands. This outbreak might cause anomalies in the dataset. The data is made available through the AeroDataBox API, on [4]. The data contains a seasonal pattern on a weekly level.

The *inventory high* and *inventory low* use cases both describe daily inventory levels of bicycles at shops in the Netherlands. The first has a high average inventory level, and the second has a low average inventory level. Constructing this data requires communication between various systems, as the shops are not necessarily of the same owner. This might be the cause of missing or duplicate data. The set is made available confidentially for this research.

The *sales* use case describes daily sales data of a product used in the flow control of industrial pipelines. Often, sales arise in batches. However, they are difficult to separate from regular orders as they might not be classified properly. This dataset was also made available confidentially for this research. A weekly seasonal pattern is visible in the data.

The *taxi* use case describes the number of taxi trips made in New York City on an hourly level. This source originates from the NAB. As indicated in their documentation, special events like the NYC marathon or a snowstorm might cause anomalies. The data shows both a daily and weekly seasonal pattern.

The *traffic* use case describes the average speed of vehicles passing over a segment of Dutch highway. These statistics are publicly available on the Nationale Databank Wegverkeersgegevens (NDW) website through their expert module open data on [92]. We average the statistics on an hourly level. Traffic jams, accidents, or construction might be the cause of anomalies. The data contains a daily pattern.

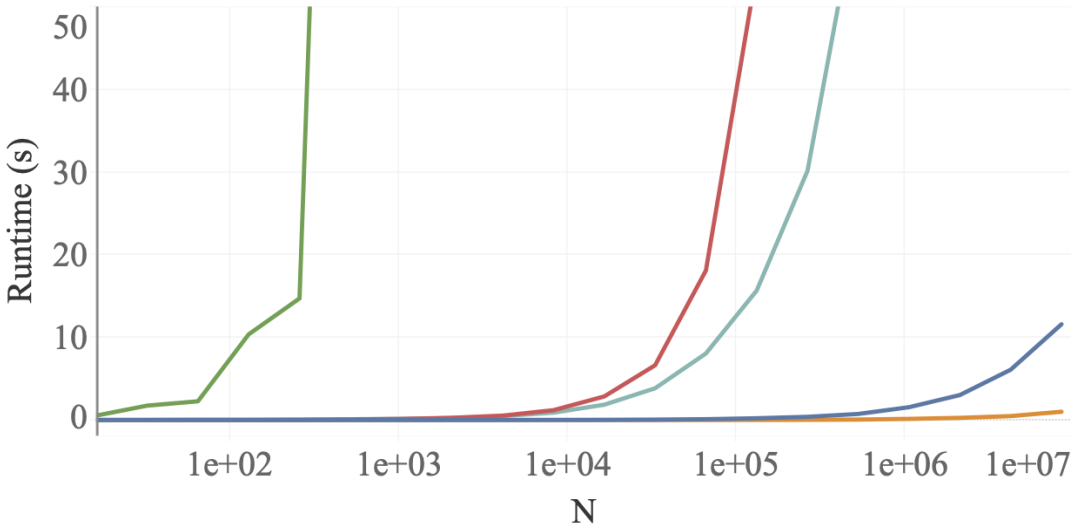


FIGURE 2.5: Runtime (seconds) vs time series length (N). From left to right: tsoutlier (green), KNN-CAD (red), PEWMA (light blue), custom (blue), custom without seasonality (orange).

The *twitter* use case describes the number of mentions of the corporation Apple on Twitter. The source of this dataset is the NAB, again. Anomalies might be caused by various events centered around the company. As in most use-cases, the data shows a daily seasonal pattern.

2.6 Results

Table 2.2 compares the performance of all methods on all use cases. We observe large differences in both dimensions. Generally, our proposed method outperforms the others with an average optimized F1-score of 0.93. PEWMA comes second with a score of 0.77, followed by tsoutlier (0.71) and KNN-CAD (0.33). In six of the nine use cases, our methodology achieves the highest score. Additionally, its minimum F1-score is 0.75, which is much larger than the second-best minimum F1-score, which is PEWMA with a score of 0.24. Looking at the default parameter settings, the minimum F1-score drops to 0.29, however, for all other methodologies the minimum score on default parameters equals 0. The custom, tsoutlier, and PEWMA methods achieve a maximum F1-score of 1, however, KNN-CAD achieves a maximum of 0.57.

Table 2.3 compares the runtime of all methods on all use cases. It displays both the average and standard deviation over multiple runs. We observe our proposed methodology is the fastest on each use case. In total, it is approximately a factor 10 faster than PEWMA, a factor 100 than KNN-CAD, and a factor 100,000 than tsoutlier. The runtime of tsoutlier is highly dependent on fitting the ARIMA model, which typically shows challenges in seasonal times series such as *traffic* or *flights*.

Figure 2.5 visualizes the runtime of all methods as a function of the time series length. We again observe large differences amongst the methods. The tsoutlier method is slowest, KNN-CAD and PEWMA are relatively close, and our custom method is fastest. We implemented two versions of our approach: one with and one

TABLE 2.3: Runtime (seconds) compared by method and use case: average and standard deviation. Averaged over 1000 trials, except tsoutlier, which is averaged over 10 trails.

use case	tsoutlier	PEWMA	KNN-CAD	custom
api	$5.2e+01 \pm 2.2e-01$	$5.8e-02 \pm 2.0e-02$	$1.4e-01 \pm 8.9e-03$	$4.5e-04 \pm 8.7e-04$
taxi	$1.4e+02 \pm 1.8e-01$	$5.8e-02 \pm 2.1e-02$	$1.4e-01 \pm 8.9e-03$	$9.9e-03 \pm 3.7e-03$
sales	$1.3e+01 \pm 1.1e-01$	$5.6e-02 \pm 1.4e-02$	$1.4e-01 \pm 1.8e-02$	$3.4e-03 \pm 2.0e-03$
traffic	$6.9e+02 \pm 8.8e-01$	$5.8e-02 \pm 2.1e-02$	$1.4e-01 \pm 9.1e-03$	$3.4e-03 \pm 1.7e-03$
flights	$2.0e+02 \pm 9.8e-01$	$3.9e-02 \pm 1.4e-02$	$9.4e-02 \pm 1.5e-02$	$3.2e-03 \pm 1.7e-03$
twitter	$4.7e+01 \pm 2.4e-01$	$5.9e-02 \pm 2.2e-02$	$1.4e-01 \pm 9.3e-03$	$3.5e-03 \pm 2.0e-03$
ad exchange	$3.4e+01 \pm 2.1e+00$	$5.3e-02 \pm 1.4e-02$	$1.3e-01 \pm 2.0e-02$	$3.8e-03 \pm 2.7e-03$
inventory low	$1.9e+00 \pm 1.4e-02$	$6.0e-02 \pm 1.6e-02$	$1.5e-01 \pm 1.9e-02$	$4.3e-04 \pm 4.3e-04$
inventory high	$5.3e+01 \pm 3.8e-01$	$6.0e-02 \pm 1.6e-02$	$1.4e-01 \pm 1.6e-02$	$4.2e-04 \pm 7.4e-04$
sum	$1.2e+03 \pm 6.9e-01$	$5.0e-01 \pm 3.2e-04$	$1.2e+00 \pm 2.1e-04$	$2.9e-02 \pm 4.0e-06$

without the correction for seasonal effects. The latter version by far outperforms the others, with a runtime of one second on a series of length $1e+07$.

Figure 2.6 visualizes the impact of the parameter β on the classification of use case *flights*. Two extremes are used: 0 and 1. We observe that the parameter has the effect we expected; setting β to 1 steers the classification towards outliers having a large local effect and setting β to 0 steers the classification towards outliers having a large global effect. The experts only classified the outlier at $t = 170$, which both parameter settings classify correctly.

Figure 2.7 visualizes the classification of KNN-CAD and PEWMA using default parameters on use case *inventory low*. Both methods achieve low F1 scores, 0 and 0.2, respectively. KNN-CAD only generates false positives and is not able to classify the two outliers at timestamps $t = 359$ and $t = 374$. PEWMA does identify both outliers, however, generates many false positives. Remarkably, both methods classify an outlier whilst the value of the series equals its value at the previous timestamp, at $t = 257$ and $t = 275$.

2.7 Discussion

In the introduction, we claimed our proposed methodology has several advantages: it is intuitive, model-free, accurate, robust, and computationally inexpensive. Throughout this chapter, we have substantiated each of these properties. It is intuitive, as its definition is relatively simple and classified outliers are, as a result, explainable. It is model-free, as the seasonal component is optional. Its accuracy is highlighted in an optimized F1-score of 0.93, by far outperforming the second-best method, having a score of 0.77. Its robustness is displayed in a minimum F1-score of 0.75, outperforming the second-best method, having a minimum score of 0.24. Computationally it is inexpensive, adding up to a 100,000 fold performance gain on real-life use cases, and it is able to classify a fictional time series of one million data points within two seconds.

Two mentioned drawbacks of the methodology are its scope towards additive outliers only and its imposed parameters. The parameters are reduced to a minimum, however, consisting of α (sensitivity) and β (balance local vs global). The scope towards additive outliers is currently the largest limiting factor. However, it might be possible to detect more outlier types following a similar logic as proposed in

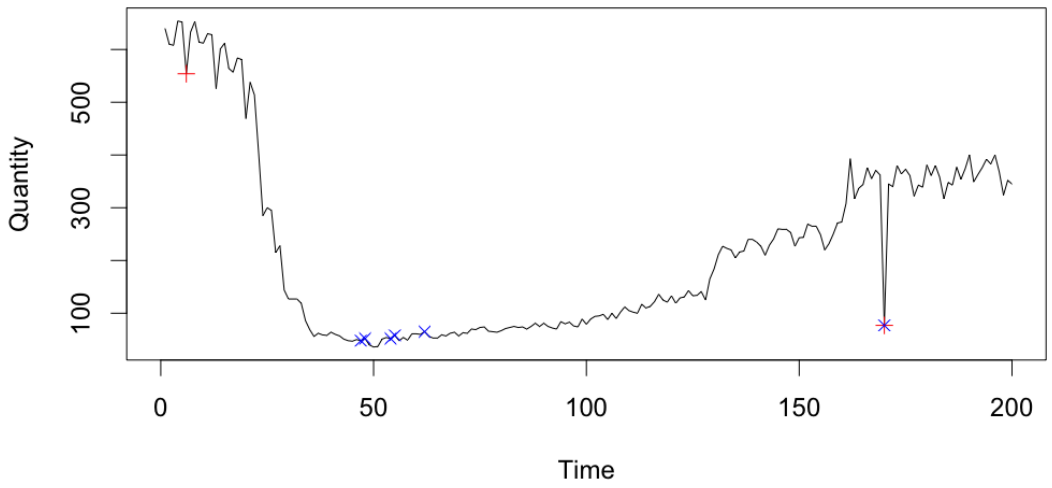


FIGURE 2.6: The influence of parameter β: detection on use case *flights* using two parameter settings: $\alpha = 2$ and $\beta = 0$ (red, plus) and $\alpha = 2$ and $\beta = 1$ (blue, cross).

this chapter. For example, level shifts might be considered 'a large change not preceded and followed by a large change'. Further investigation is necessary to evaluate whether this is a solid approach.

As a result of its computational inexpensiveness, the methodology is highly suitable for big data sets. Nowadays, data is being generated by an increasing number of sources. For example, sensors can generate data on a millisecond level. Analyzing these data requires methodology that can process the data at least faster than it is being generated. Additionally, a sliding window-based approach using a running median can be implemented to make the methodology applicable to data streams.

The current methodology can be used for detecting consecutive outliers by adjusting the input data. For example, if the time series consists of daily data, and we want to check if a certain week is abnormally low or high, we can simply aggregate the data before inserting it in the methodology.

Additionally, we could use our methodology for smoothing noisy data. As we evaluate and assign a score to each individual data point in the series, we could use this score for smoothing. This can be done by taking a weighted average of the surrounding values, weighted inversely proportionate to the outlier score. Setting α to a low value will consider each surrounding data point using a different weight.

The evaluation of methodologies is challenging, as different methods and experts might give different results. We chose to tackle most of the issues by asking multiple experts and through a majority vote establishing a ground truth. However, the challenge that not all methods classify similar outlier types remains. Our methodology is scoped at additive outliers, whereas some benchmarks have a broader scope. Therefore, it is expected that our methodology scores slightly higher than some benchmarks. However, the observed difference in both the F1 score and runtime displays the relevancy of our proposed methodology.

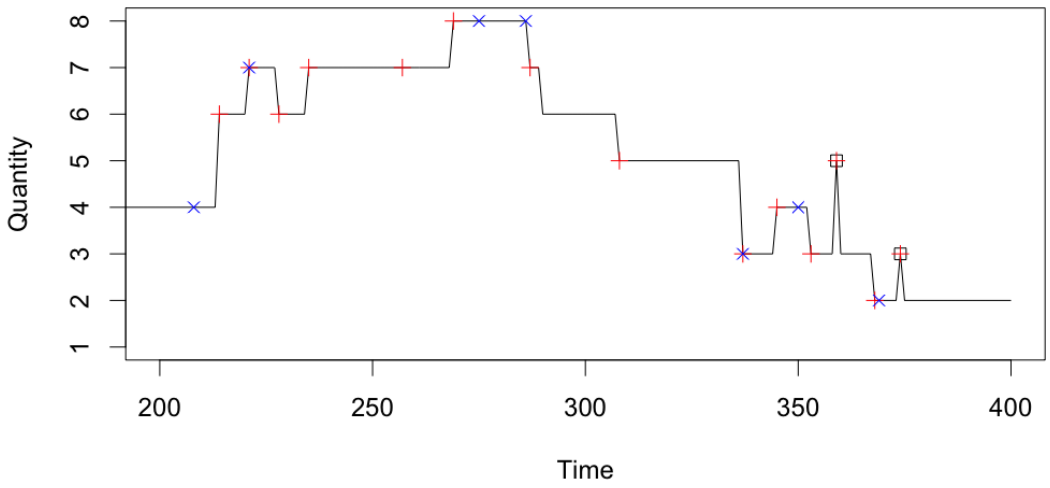


FIGURE 2.7: Classification of experts, KNN-CAD, and PEWMA on the last 200 timestamps of use case *inventory low*. Classified outliers are marked per method: expert (black, square), KNN-CAD (blue, cross), and PEWMA (red, plus).

2.8 Implementation in Practice

The algorithm presented in this section has been implemented in practice at Gazelle, a Dutch bicycle manufacturer. The algorithm is being used to detect outliers in Point of Sales (PoS) data, which consists of sales and inventory data streams. These streams are volatile, as hundreds of sales locations share their data at different times throughout the day. Additionally, the received data might be incomplete or duplicate.

Feedback from the implementation was positive, as illustrated by the quote below:

> Fedde Wildenbeest (CFO) Gazelle: Jesper heeft met zijn PhD project bij Gazelle een enorme bijdrage geleverd aan de beschikbaarheid en het gebruik van goede doorverkoop en voorraadinformatie uit de markt. De data uit de PoS systemen was al langer beschikbaar bij Gazelle, door allerlei problemen met dubbelingen, ontbrekende data vanuit dealers etcetera was deze echter niet betrouwbaar genoeg om als stuurinformatie voor verkoop- en productieplanning te gebruiken. Na het afronden van zijn opdracht waarbij o.a. cleaning, controle en ontdubbelingsmechanismen zijn ingebouwd is deze informatie een geïntegreerd onderdeel van de managementinformatie bij Gazelle geworden en bijvoorbeeld standaard onderdeel van het MT overleg.

Acknowledgments

The authors want to thank Rik van Leeuwen for providing the data underlying the use case *api* and Robin van Ruitenbeek for providing the data underlying the use case *sales*.

3 Understanding Human Mobility for Data-Driven Policy Making

3.1 Summary

This study aims to identify the patterns of behavior which underlie human mobility. More specifically, we compare commuters who drive in a car with those who use the train in the same geographic region of the Netherlands. We try to understand the mode choices of the commuters based on three factors: the cost of the transport mode, the CO_2 emissions, and the travel time. The analysis has been based on data consisting of travel transactions in the Netherlands during 2018 containing over half a million records. We show how this raw data can be transformed into relevant insights on the three factors. The results can be used to stimulate behavioral change proactively. Moreover, the data and results can also be utilized to improve trip planners.

3.2 Introduction

Commuting long times and distances has become a regular part of the daily routine for most people. How people travel to work is in part a function of personal preference, which has been discussed in terms of comfort in the vehicle, addressing issues such as temperature, air quality, noise, vibration, light, and ergonomics [29]. However, the mode choice also reflects contextual factors [110], including economics – the cost and acceptability of different commuting modes due to travel times and CO_2 emissions.

The continued expansion of commuting distance and time in cars has obvious environmental consequences as it relies on fossil fuels. Pollution generated by cars has health consequences for travelers [41, 146]. Commuting can also be stressful, and the duration of the trip contributes to the stress experienced by workers [36, 147]. There are few studies that have looked at commuting experiences for mass transit commuters. Singer et al. [122] found increased stress on crowded trains. Indices of stress were reduced when train commutes were improved by route changes that shortened commuting time and enhanced predictability of the trip [147].

Few studies have directly compared riders across modalities, such as train versus car commuting. Based on available information, one might predict that car commuting is less preferable than train commuting, particularly because of differences in predictability and effort, both of which have been linked to stress [37, 74]. For example, the vagaries of traffic and sudden onset of accidents or other kinds of traffic jams make driving times for the commute to and from work unpredictable, especially in densely populated major metropolitan areas. Driving also requires constant

TABLE 3.1: Mobility transactions dataset, sampled and containing fictive values because of data privacy agreements.

type	start_ts	end_ts	start_city	end_city	distance	duration	CO_2
car	10/03/2018 07:45	10/03/2018 08:24	Utrecht	Wageningen	49	39	7.02
train	28/11/2018 08:32	28/11/2018 09:20	Zandvoort	Amsterdam	27	48	0.16
train	09/04/2018 16:51	09/04/2018 17:37	Amersfoort	Zwolle	66	46	0.4
car	09/07/2018 14:00	09/07/2018 14:36	Beilen	Groningen	52	36	8.71
car	21/01/2018 07:45	21/01/2018 08:33	Wijchen	Den Bosch	39	48	6.82

attention and effort – more so as conditions worsen. Trains are likely to be more predictable and less effortful as a mode of travel.

On the other hand, driving may afford a higher level of control for the driver. The driver has more ability to influence the time of departure, route, and road speed. Williams et al. [151] found that drivers in the UK had higher levels of perceived control than those using other transit modes. Past research in other situations also indicates that control may be an important factor in mode choice [45]. Car commuting affords a higher degree of control over social interaction, a critical aspect of privacy. Indeed, if drivers do have higher levels of perceived control than do train commuters, driving may be a more preferred mode of travel.

In this chapter, we compare commuters who drive in a car with those who use the train in the same geographic region of the Netherlands. We try to understand the mode choices of the commuters based on three factors: the cost of the transport mode, the CO_2 emissions, and the travel time. For this purpose, we use a rich dataset of mobility transactions by employees of a private company. We show how to transform the data into relevant insights, such as congestion, to calculate the three above-stated factors. This allows us to compute relevant statistics by predicting travel mode choice. This predictive model, in turn, can be used for policy making and better network decisions.

3.3 Problem Formulation

The focus of this study is on understanding human travel behavior when it comes to using the car and the train as travel modalities. Our approach is to compare how the modalities train and car differ in terms of CO_2 emissions, cost, and travel time.

We analyze two rich datasets: mobility transactions and highway sensors. Both collect data based on real-life events through human interaction. The first challenge is that the data cannot be used directly for analysis. We show that one should carefully transform the data to avoid biases in the analysis. Another challenge is to quantify congestion when traveling from A to B, based solely on highway sensors. We develop a methodology to address both these challenges.

After we have transformed the data, we provide an analysis on the differences in human behavior to explain why a modality choice between train and car is made. This result can be used to predict modality choices, and we show how they can be used in practice through a use case.

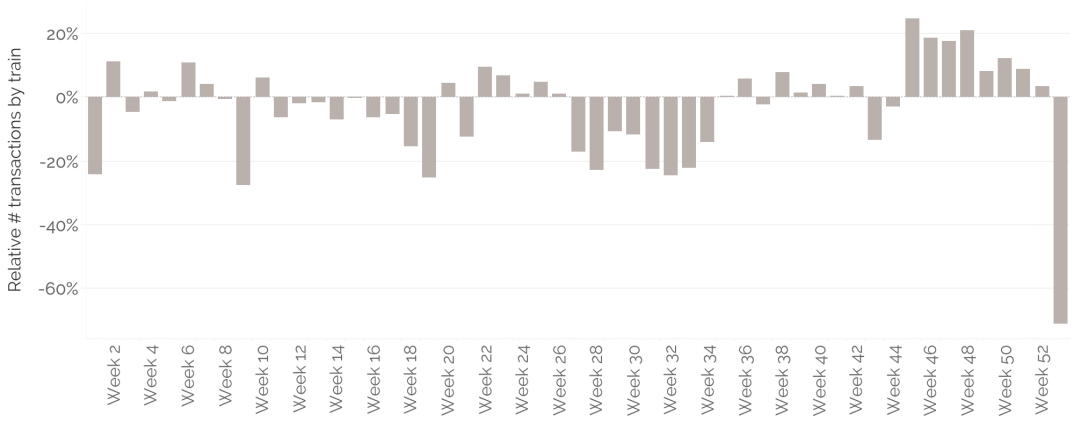


FIGURE 3.1: Usage of the train throughout the weeks of the year, relative to the average number of transactions per week.

3.4 Methodology

We present various methodologies for answering our research questions. First, we analyze a rich dataset containing mobility transactions. Hereafter, we explain how to process and combine statistics related to congestion.

3.4.1 Mobility Transactions

We first analyze a mobility data set that is unique in its kind. It has been made available for analysis under strict conditions by a private company providing mobility to its customers through a mobility card. The data contains mobility transactions that are registered through automated systems. In this section, we describe its origin, show how to process such data, and present various statistics originating from an exploratory analysis.

As in [130], the full dataset contains over half a million mobility transactions from over a thousand employees originating from various companies and offices in the Netherlands. The data analyzed concerns a period of one entire year 2018. Amongst other statistics, we know the transport type, start and end date and time, start and end location, distance, duration, and costs of each transaction. In this chapter, we focus on a processed dataset containing trips with transport types 'car' and 'public transport' only. Table 3.1 shows a representative sample of the most important columns in the dataset, containing fictive values because of data privacy agreements. Each record in Table 3.1 shows a mobility transaction. The 'type' specifies the modality used to travel: car or train. The transaction starts at timestamp 'start_ts' and ends at timestamp 'end_ts'. The starting location is given by 'start_city' and the destination location by 'end_city'. The last three columns display the statistics on this transaction: the distance measured in kilometers ('distance'), the duration in minutes ('duration'), and the CO_2 emissions measured in kilograms ('CO_2'). The CO_2 emissions are estimated through our own analysis, which is described in the last paragraph of this section.

The raw data needs to be processed before it can be used to answer our research question. Most importantly, we need to apply the appropriate filters. As the data is gathered through automated systems, it contains transactions that we do not wish to analyze. First, we filter out short trips. Most car trips are short, as a new trip

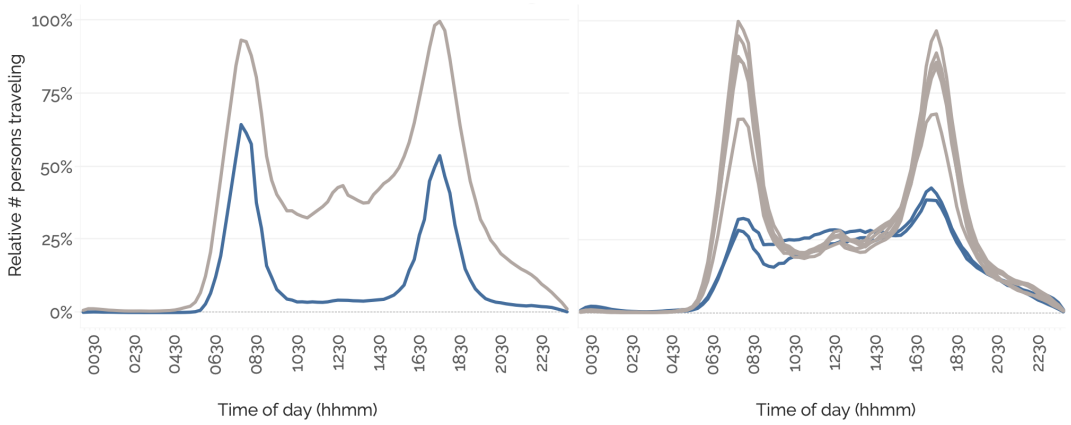


FIGURE 3.2: People traveling simultaneously: (a) split by car (grey) and train (blue); (b) split by weekdays and colored by working days (grey) and weekend (blue).

is registered each time the engine is turned on or off. Also, for public transport, in the Netherlands, some stations cannot be traversed without checking in and out at each entrance. Thus, trips that have a duration (in time) shorter than a threshold are filtered. Second, we filter trips having a highly similar start and end location. These trips are difficult to analyze, as it is challenging to determine the true destination or purpose of the trip. Third, we filter car trips starting and ending at gas stations. We do not see these trips as the intended start or end locations of the users. They are forced to visit gas stations in order to reach their destination. Also, some gas stations are not accessible by public transport. In addition to filtering, we apply a correction on the start and end locations of trips by public transport. The raw locations will always be at stations. However, these are not the actual start and end locations of the travel. We correct these locations by sampling a random address within the area that is reachable within ten minutes by bike. Afterward, we re-compute the travel time and distance using an API.

After gathering and processing the raw data, we can explore the data. We start by looking at the usage of the train. Figure 3.1 shows the relationship between the week of the year and the relative number of transactions by train. The percentage is relative to the mean number of train transactions per week. A clear relationship can be observed between the train usage and the weeks containing holidays. In the first week of the year, the train usage is at a low level of -22%. The weeks containing the spring holidays, summer holidays, and the Christmas holiday all show a train usage lower than -20%. Interestingly, around November and at the beginning of December, the train usage is relatively high. This might be explained by poor weather conditions or a relatively low number of holidays during these weeks.

Next, we take a closer look at the time of the day at which people travel. For each transaction in our dataset, we know the timestamp of the start and end. Therefore, we can derive the number of people that are traveling at any minute of any day. Figure 3.2 shows the results of this exercise. On the left (a), it shows the relative number of people traveling during the day split by car (grey) and train (blue). On the right (b), it shows the same information split by days during the week (grey) and days during the weekend (blue). The numbers are relative to the maximum. In both graphs, high peaks can be observed during typical commuting hours. However, clear differences are visible between trips by car and train. A much larger

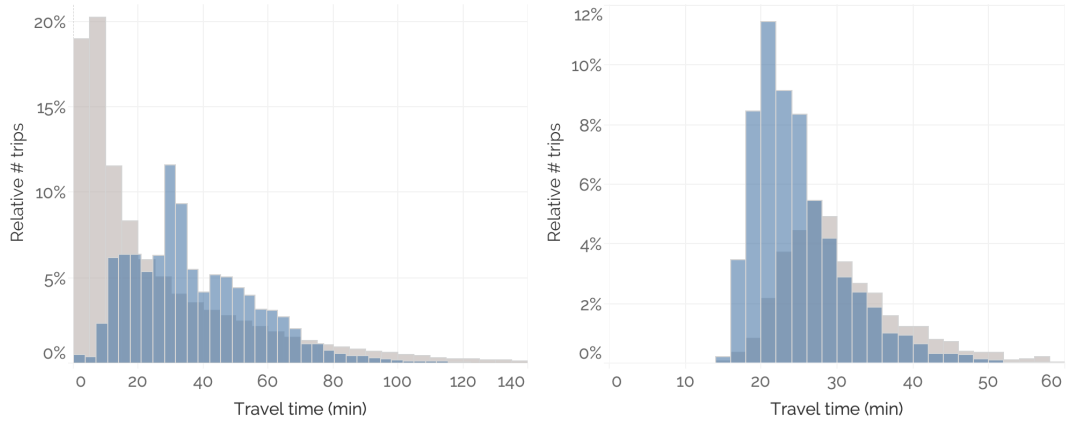


FIGURE 3.3: Distributions of travel duration split by car (grey) and train (blue): (a) amongst all transactions; (b) amongst transactions between Utrecht and Amersfoort.

number of people are traveling during the middle of the day, and a small peak can be observed after lunch. Besides the graph split by train and car, large differences are visible amongst days in the week. On the weekend, fewer people travel during commute hours. Next to these differences, the number of people traveling is the largest on Tuesday, the smallest on Sunday and during the workweek the lowest on Friday.

Figure 3.3 compares the distribution of travel duration for car and train transactions. On the left (a), we see that cars are more frequently used for relatively short trips. In contrast, trains are generally used for relatively long trips. This could be explained by the fact that the car might be faster. However, the right graph (b) seems to reject this hypothesis. The travel time distribution for both train and car between two specific cities in the Netherlands is shown here. The cities are Utrecht and Amersfoort, both located in the center of the Netherlands. Between them, the travel time distribution of the train is smaller than that of the car.

To further investigate the interaction between transport type and speed, we explore two relations: that between trip length and speed; and that between trip length and the frequency of occurrence. Figure 3.4 shows these graphs, for both car (grey) and train (blue). If we want to approach answering the question of which transport type is fastest, we need to consider both. On the left, we overall see an increasing relation between trip length and average speed. Also, the average speed of the train always lies below that of the car. On the right, we see a different distribution of trip length for both transport types. The car is often used for relatively short trips, whereas the train is used for relatively long trips. Thus, if we would simply compare the average speed of all car transactions with all train transactions, we would get a biased result.

Lastly, we combine transactions in our dataset to estimate the carbon footprint in terms of CO_2 emissions. Different transport types have different carbon footprints. Regarding public transport, these figures are publicly available. In the Netherlands, through [81]. However, for car transactions, these figures depend on a range of factors, such as the engine, driving behavior, car weight, or outside temperature. This makes it more difficult to quantify the footprint. However, the full dataset contains fuel-related transactions. Each re-fuel of a car is stored as a transaction,

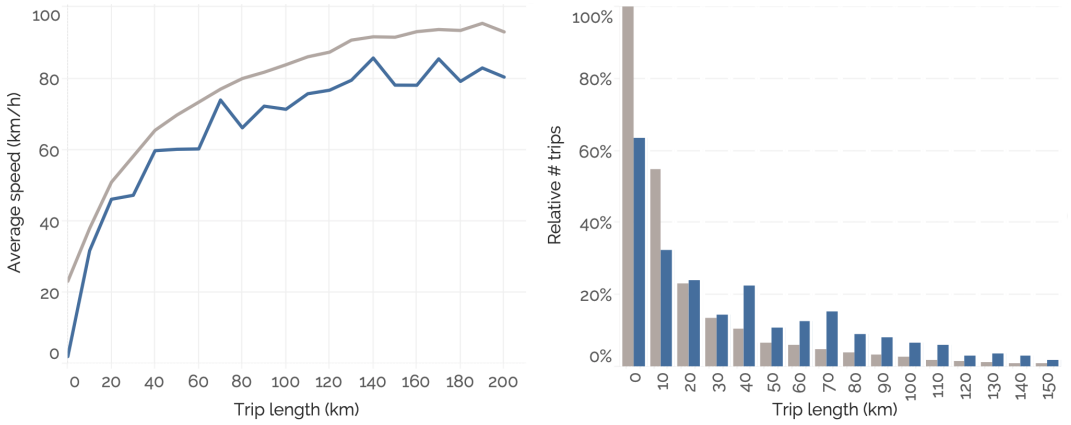


FIGURE 3.4: Interaction between transport type and speed, for car (grey) and train (blue): (a) interaction between trip length and average speed; (b) interaction between trip length and number of trips.

containing the volume and type of fuel used. Using these statistics, we can estimate the amount of fuel burned for each car transaction in the full dataset. We join this to the mobility transactions dataset, hereby creating the CO_2 column. This can be translated towards kilograms of CO_2.

3.4.2 Congestion

Congestion is a factor that affects travel time. Depending on the location of the congestion and the route, this might have a large or minor influence. Still, in [130], we have shown that overall there is a relation between the departure time and the expected travel time. Thus, if we want to make network-related decisions, we need to measure and quantify congestion. In this section, we describe how to process measurements related to congestion on roads.

In the Netherlands, the Nationale Databank Wegverkeersgegevens (NDW) tracks the speed and volume of cars by using more than 37,000 sensors on federal roads. These statistics are made available publicly through their data portal on a minute level. Figure 3.6 visualizes these measurement sites. On the left (a), most measurement sites in the Netherlands are shown. On the right (b), the sites within Amsterdam are shown. When closely studying the sites within Amsterdam, we can see these are tied to a road segment, thus, being specific for a certain direction of traffic flow. This allows us to compute statistics on both a micro and macro scale. Our goal is to quantify congestion on the road at a specific time, between two specific locations, and in a certain direction. To do so, we need to process this data appropriately.

The main challenges in processing the NDW data are finding relevant sites and determining a congestion level for each site. Finding relevant measurement sites is challenging as there are thousands of sites. However, we are only interested in those sites covering the traveled route. We filter the relevant sites by fitting a rectangle between the start and end coordinate of the corresponding trip. Depending on the trip length, the width of the rectangle is adjusted. We only consider sites positioned inside the rectangle. Next, we filter these sites on having a direction within a threshold of 90 degrees of the general trip direction. We can determine the direction of the

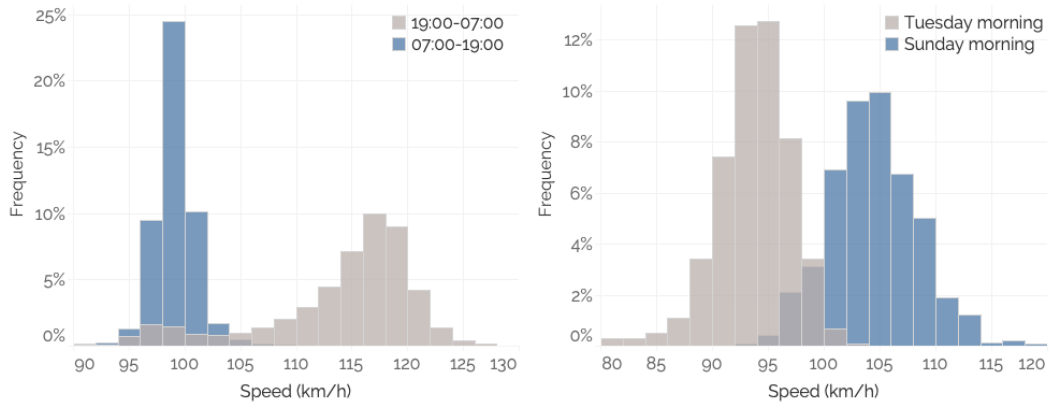


FIGURE 3.5: Differences in observed speed values depending on: (a) time of day; (b) day within week.

traffic that the site is measuring because we know the ID and location of the next site on the road segment.

Determining a congestion level for each measurement site is a challenge as well. The sites can be located at different road types with different speed limits. To further complicate this, the speed limit can vary throughout the day. Thus, we cannot directly take the velocity of the traffic as a statistic of congestion. Instead, we first derive the distribution of speed measurement values for each site. This can be dependent on the time of the day. Using these distributions, we can convert each measurement value to a congestion score by comparing it to the location of its corresponding distribution.

Figure 3.5 visualizes the distribution of observed speed values, split by two different dimensions. On the left (a), it is split by time of day, and on the right (b), it is split by day of week. Both graphs display observed values by a unique measurement site. Whilst the differences on the right graph can be explained through congestion, those on the left cannot. On the left, it is a site placed on a Dutch highway having a different speed limit during the night. This limit is enforced through trajectory speed control, so drivers are hesitant to exceed it. Remarkably, during the night, some vehicles still drive at the same speed as the one allowed during the day. This could be explained by limits on certain vehicles (e.g., heavy trucks) or due to habit. The graph on the right (b), displays the observed values on a measurement site that is sensitive to traffic jams. It compares speed values observed on Tuesday morning with those observed on Sunday morning in November. We observe that the speed values on Tuesday seem much lower than those on Sunday, which is intuitive as more people travel on Tuesday morning (Figure 3.3).

Figure 3.7 visualizes the result of normalizing the observed speed values. It compares two measurement sites in different parts of the country having a different speed limit. One has a limit of 100 km/h (grey), the other a limit of 80 km/h (blue). If a trip traverses both sites, we need to consolidate both, despite these differences. The normalized speed values seem to achieve this. We observe that during the night and the middle of the day, the normalized values are highly similar for both sites. Both are close to, or slightly above 0. This is intuitive as the observed values lie close to the speed limit. The speed values hardly exceed, but often lie underneath this limit, thus, we expect a slightly positive normalized speed. During rush hours, we

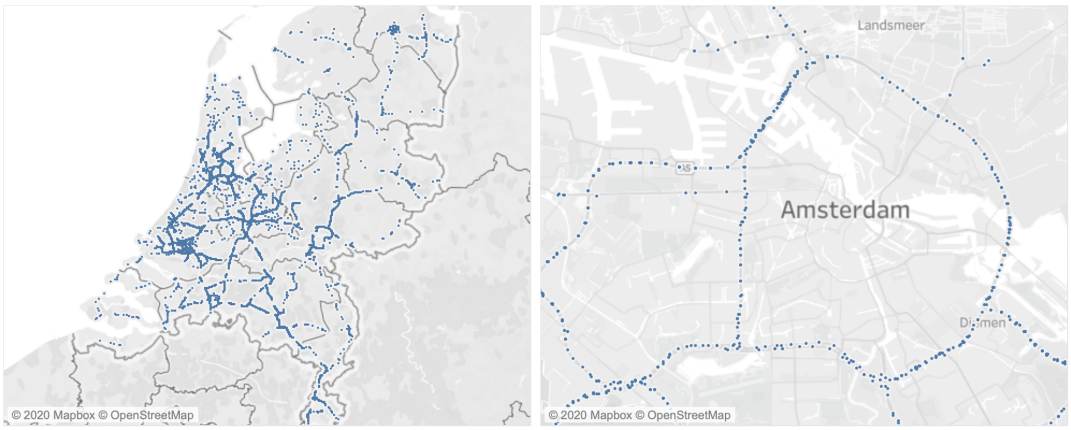


FIGURE 3.6: NDW measurement sites: (a) in the Netherlands; (b) surrounding Amsterdam.

observe some differences. Those are intuitive as well, as they directly are a result of the raw data values.

3.5 Results

This section highlights our most important findings from analyzing the mobility transactions and congestion data sets.

Figure 3.8 compares the mobility transactions of cars and the train through three different statistics: CO_2 emissions, cost, and speed. We observe a significant difference in terms of CO_2 emissions. Compared to a car, the train hardly emits CO_2. When taking into account well-to-tank emissions, this difference grows even larger. Regarding cost, the train is more expensive when looking at variable cost. However, when including fixed costs, the train has a lower cost per kilometer. Looking at speed, we observe that both transport types are relatively close. The car is slightly faster than the train. The differences during rush hour are most prominent. The speed of the car decreases during rush hour, whereas the speed of the train increases. This is likely explained by congested roads and a higher number of trains scheduled during rush hour.

Remarkably, the total cost (variable + flexible) of the car is twice as high as the cost of the train. The variable cost is the leading cause, consisting of more than €0.30 per kilometer. When making a fair comparison between car and train, we think both factors should be taken into account. Besides, the observed difference in speed hardly changes looking at both transport types. This might be because we average over the whole country, so local differences might be larger. The observed difference in CO_2 is extreme, however, it conforms with our expectations.

Figure 3.9 shows the result of computing the congestion between two locations. It compares the normalized speed (low speed indicates high congestion) amongst the hours within a day. All days in 2018 are averaged in making this graph. The measurement sites taken into account lie between two cities in the Netherlands: Amsterdam and Almere. The normalized speed is computed in both directions, from Amsterdam to Almere (blue) and from Almere to Amsterdam (grey). We focus on these cities because a large part of the inhabitants of Almere work in and commute to Amsterdam. This effect is visible in the computed congestion. Traffic heading

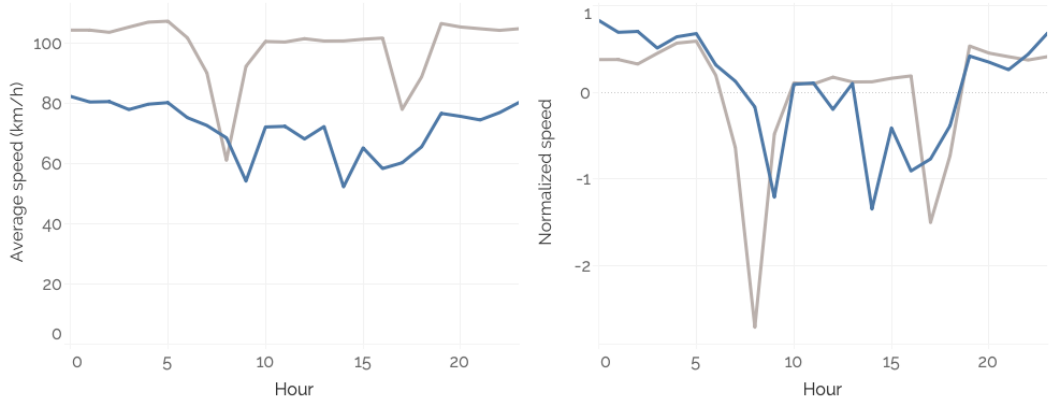


FIGURE 3.7: Observed (a) vs normalized (b) speed values by measurement sites having a speed limit of 100 km/h (grey) and 80 km/h (blue).

towards Amsterdam is congested during both morning and evening rush hours, whereas traffic heading towards Almere is only congested during the evening rush hours.

3.6 Use Cases

The data analyzed in this research and the corresponding results can be used for predicting the modality choice of individuals. Understanding the relation between CO_2, cost, and time and modality choice allows us to do so accurately. In [130], we did so with a 97% accuracy. This is largely based on the same mobility transactions dataset, enriched with a generic dataset regarding reachability features, which quantify how well the network of a modality is developed. Interestingly, the main predictors of this model are the reachability features, more so than specific travel times. Additionally, the travel type (commute or personal) showed to have a large influence on travel mode choice. The reliable predictions of this model can help users in their decision-making. For example, we can send proactive messages to notify users of alternative travel modes, or we can increase the visibility of relevant travel modes in travel planners.

If the user allows us, we can notify him/her of alternative travel modes. This can be relevant because users might lack the knowledge, construction or traffic jams are anticipated, or because of policymakers wanting to stimulate behavioral change. In all cases, we only want to send notifications to users with a certain probability of adjusting their behavior. For instance, if a company wants to stimulate train usage to its employees, they could notify all people within a certain distance or travel time from their respective office. However, this ignores the relation between other modalities. It could be that for an individual, the travel time using the train is 25 minutes and ten minutes by using the car. These users will have a relatively low probability of traveling by train. On the other hand, there might be users having 40 minutes of travel time using the train and 30 minutes using the car. The second group of users would have a higher probability of traveling by train. The model estimates these probabilities, including more statistics than travel time, and helps select relevant users for behavioral change.

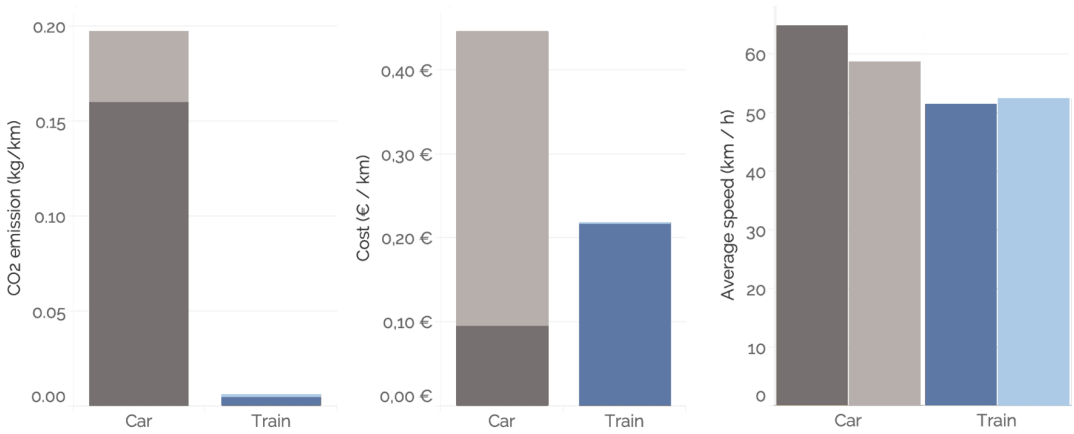


FIGURE 3.8: Comparison of car (grey) and train (blue) through: (a) CO_2 emission by tank-to-wheel (dark) and well-to-tank (light); (b) cost by variable (dark) and fixed (light); (c) speed outside (dark) and during (light) rush hour.

Besides proactively stimulating behavioral change, we can use our data and results to improve trip planners. These face the challenge of displaying the most relevant modalities to their users. Using our insights, we can adjust the visibility of travel modes based on the estimated probability they will be chosen. Currently, we can balance train and car. In the future, we can extend the analysis to include more forms of mobility by including shared concepts such as bikes, scooters, or cars.

3.7 Conclusion

In this research, we have developed methods for handling data sets containing mobility transactions and congestion. In our opinion, we have shown promising results for different use cases. Still, our methodology can be further improved. In this section, we further discuss our findings and highlight potential improvements to our methodology.

Our results comparing train and car mobility concerning CO_2, cost, and speed require some side notes. First of all, we assume car trips are executed with one person at a time. We could apply a general correction, however, we have little data to make an educated guess. Therefore, we left the statistics as is. Regarding the CO_2 emissions, all our transactions in the dataset are based on data from 2018. Given the electrification in the automotive industry, we expect this to impact the emissions. Tank-to-wheel emissions might decrease, however, the emissions due to producing an electric car might increase because of the battery production. Finally, we realize that the historic data introduces a bias regarding speed. For example, transactions that would take an extremely long time with public transport might not be executed, hence not showing up in our dataset, thereby not influencing public transport speed in our analysis.

We see the most significant potential for improvement in the methodology to measure congestion. This can be done in both selecting relevant measurement sites and in better interpreting the speed values coming from them. We can improve on selecting sites between an origin and destination by integrating a routing API to give us exact routes between them instead of fitting a rectangle. Having these routes, we

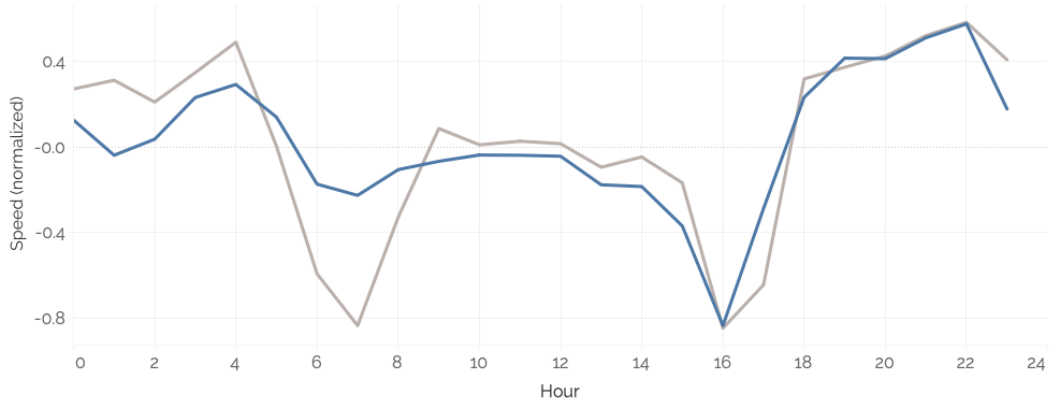


FIGURE 3.9: Quantifying congestion: congestion from Amsterdam to Almere (blue) and from Almere to Amsterdam (grey) split by hour of the day.

can focus only on the sites covering them. As a result of this, we would reduce the number of measurement sites we take into account.

Further, we can better interpret the speed values resulting from each measurement site by considering the speed limit on the corresponding road segment. Currently, we implicitly derive this speed limit by analyzing the distribution of observations from a specific measurement site. However, if a road segment is often heavily congested, this might influence our derived limit.

Besides these methodological improvements, it would be relevant to incorporate data on more modalities than the train and the car. The mobility transactions dataset already contains more modalities, however, these volumes are too low to draw conclusions. For example, its structure is set up also to incorporate trips done by shared scooters or shared cars. As these services become more common, we might be able to observe a behavioral change in some scenarios towards these modalities. Additionally, the Netherlands is well known for its usage of bicycles. Little data is generated on those, as they do not contain sensors and it hereby requires manual effort to register when and where these trips are made. Given the electrification in the bicycle industry, we do expect more data to be generated in the future. Electric bicycles can generate data through sensors, such as their motor, lights, lock, or anti-theft location modules.

4 On the Relation between Covid-19, Mobility, and the Stock Market

4.1 Summary

The Covid-19 pandemic has brought forth a major landscape shock in the mobility sector. Due to its recentness, researchers have just started studying and understanding the implications of this crisis on mobility. We contribute by combining mobility data from various sources to bring a novel angle to understanding mobility patterns during Covid-19. The goal is to expose relations between mobility and Covid-19 variables and understand them by using our data. This is crucial information for governments to understand and address the underlying root causes of the impact.

Introduction

One of the first visible impacts of the Covid-19 crisis was on transport, travel, and mobility. Mobility explained a substantial proportion of variance in transmissibility [93]. The travel restrictions adopted to limit the spread of the disease led to drastic reductions in travel and traffic. This had various implications. The disruption in the flow of goods had severe economic consequences. The measures on mobility, traffic, and transport also had a substantial impact on the socio-economic sector.

The crisis has affected all forms of transport, from bicycles, cars, public transport, maritime vessels, trains, and air flights [10]. The activity on global road transport was almost 50% below the 2019 average by the end of March. Similarly, commercial flights were almost 75% below by mid-April 2020 [68], while the global flight network density reduced by 51% [137]. Therefore, a key question is how changes in transport behavior affect each other due to Covid-19 and how they relate to the economic progression worldwide.

In the recent past, there have been a number of crises that have caused major changes in mobility patterns. For example, the Severe Acute Respiratory Syndrome (SARS) crisis of 2003 significantly affected air traffic in specific regions. The volume of traffic dropped by 35% [66]. Also, the non-essential trips with public transport dropped by 50% during the peak of the pandemic [145]. It took almost four months for the passenger numbers to return to pre-crisis levels.

The Avian Flu outbreaks of 2005 and 2013 and the Middle Eastern Respiratory Syndrome in 2015 also significantly impacted mobility. The demand for air travel in

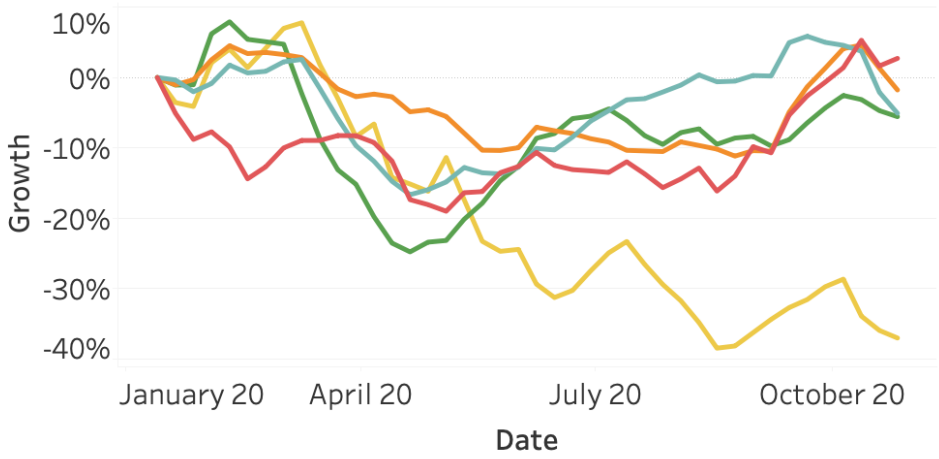


FIGURE 4.1: Vessel activity as of January 2020, split per continent. Port enters and exits are included in the data to account for inter-continental ships, exiting one continent and entering another. North America (orange), South America (yellow), Asia (red), Europe (light blue), and Oceania (green).

these cases returned back relatively quickly [66]. It is reasonable to assume that mobility patterns of the Covid-19 disease will be more in line with the patterns of SARS. The Covid-19 and SARS pandemics share the scale of the impact and the perceived risks of contagion, which are more significant than other recent pandemics.

From a behavioral perspective, it is interesting to study the patterns in mobility and the short- and long-term effects. This is crucial information for policymakers to understand and address the underlying root causes of the impact. For example, after the terrorist attacks on 9/11, there was a drop in air traffic demand that lasted five years after the attacks [28]. Studies contribute this to the risks and inconvenience of flying after new security precautions were introduced.

The Covid-19 crisis could bring forth different changes than other crises in the past. Business travel could be replaced by more video conferencing, since the technology has rapidly matured in a short period of time [50]. Reduction of demand for particular modes of transport could become permanent due to perceived risk [141]. A model shift could happen to modes of transport that avoid contact with people to have less perceived exposure to the virus [1]. Thus, a model shift could happen from public transport to bicycles [111]. Governments can use this information in change campaigns to change public behavior. This can influence which transport behaviors are more permanent after the crisis.

In this chapter, we combine mobility data from various sources to bring a novel angle to understanding mobility patterns during Covid-19. We look at mobility data from bicycles, maritime vessels, trains, car traffic, and air flights. First, we look at patterns in between these modalities. Second, we relate the patterns to the Covid-19 cases and measures, as well as the stock market. The goal is to expose relations between mobility and Covid-19 variables and understand them by using our data.

The rest of this chapter is structured as follows. In Section 4.2, we explain how we obtained the data from the various data sources. We discuss our methodology for the analysis of the data in Section 4.3. The results of the analysis are presented in

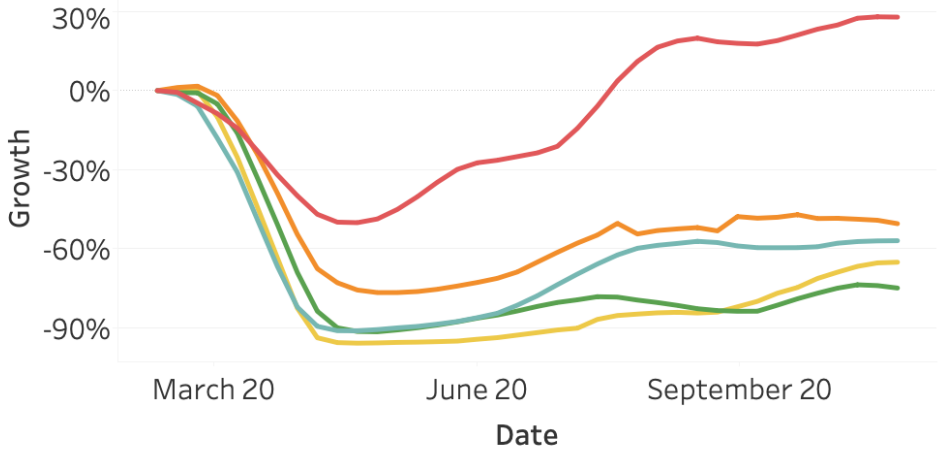


FIGURE 4.2: Flights activity as of March 2020, split per continent. North America (orange), South America (yellow), Asia (red), Europe (light blue), and Oceania (green).

Section 4.4. We conclude the chapter in Section 4.5 with a discussion on our research.

4.2 Data

We gather data from various sources to answer our research questions. In this section, we describe each source and give an initial insight into its contents. The sources are related to the usage of mobility, economic indicators, and Covid-19 statistics. All sources are on a global scale, covering countries on all continents except Antarctica. The mobility types we consider are vessels, flights, vehicles, trains, and bicycles. The economic indicators are extracted from various stock markets and Covid-19 data from official numbers by the corresponding countries. The sources are available on a daily level. However, to account for intra-week seasonality, we aggregate all sources to a weekly level to perform our analysis. Additionally, in the current section only, most graphs show the percentage change since the first known value in the corresponding time range, on a weekly level, smoothed over four weeks. We apply smoothing to apply visual focus on the general trend and take into account the percentage change to create a fair comparison amongst the different continents. All data is made publicly available through [106, 105, 107, 104, 103, 108].

Vessels

We use the real-time ais data from AISHub [6], which we collected over a time period ranging from *April 1 2019* to *December 1 2021*. The raw data is collected with an interval of 2-3 seconds. Using more than 600 ais stations, a total of 8.5 billion records of 4.8 million unique ship IDs have been retrieved. We focus on commercial and cargo ships, including passenger ships, tankers, cargo, and fishing. We exclude ship types such as military, medical, and towing.

To minimize the size of the data, we reduce the dataset to a single record per hour and use this dataset to estimate the vessel activity. We define the *vessel activity* for

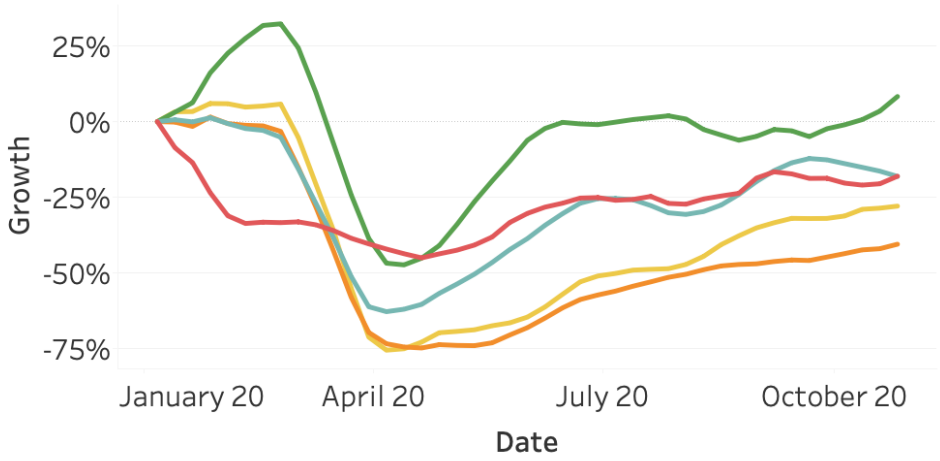


FIGURE 4.3: Vehicle activity as of January 2020, split per continent. North America (orange), South America (yellow), Asia (red), Europe (light blue), and Oceania (green).

a port in time period t as the number of vessels entering and exiting the port. To approximate if a vessel is inside a port, we assume a vessel i to be in port j if

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq \beta \tag{4.1}$$

holds. For this, (x_i, y_i) and (x_j, y_j) are the longitude and latitude pairs of both the vessel location and the center of the port. We approximate the visits using the port center coordinates of [61] instead of port shape files to speed up the computation, since it has to be performed on all records of the extensive dataset. From this, we track each vessel over time and assign a port visit if the ship has been in the port for at least $\alpha = 3$ consecutive hours. We set α to 3 as bulk carriers and oil tankers move with the lowest average speed of 24 kilometers an hour [5], indicating that at least two hours is required to pass the port without entering it for $\beta = 12$ kilometers.

To finalize the vessel activity, we solely look at the data points where a vessel is within the range of a port. From this, we assign an entrance activity if the vessel moves into the range of a new port j at time t and is detected in the same port on (or after) $t + 3$ without visiting another port in between. Therefore, if a vessel is only detected once, it is assumed to be passing the port. Similarly, we define a port exit when the vessel meets the entrance criteria and is seen in another port after $t + 3$. We assign the exit time based on the last recorded time where vessel i was seen in port j. With this, the defined vessel activity is relatively robust against sensor downtime, which frequently occurs due to vessels shutting down the system within a port.

Figure 4.1 visualizes the growth in vessel activity compared to the first week of January in 2020. We can observe large differences amongst the continents. Most notable is South America, which has generally been decreasing throughout the year. Most other continents show a dip around March and April. However, in November, they are close to their original value in January.

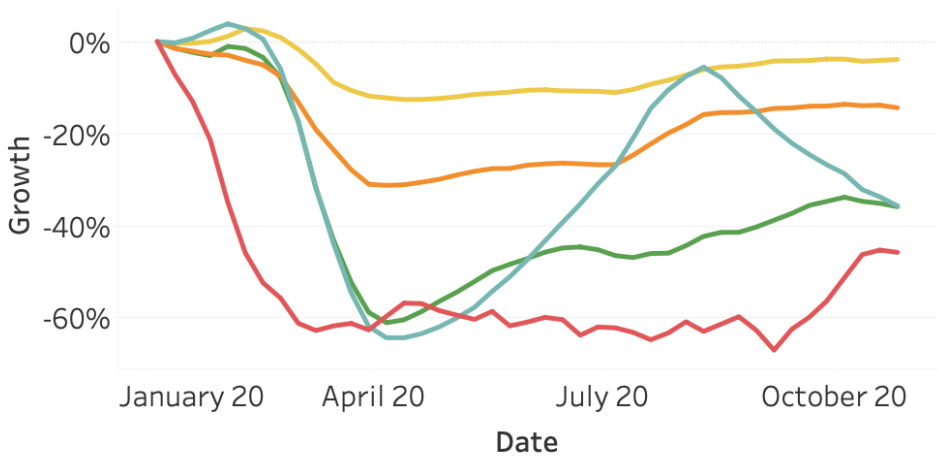


FIGURE 4.4: Online searches for train through Google as of January 2020, split per continent. North America (orange), South America (yellow), Asia (red), Europe (light blue), and Oceania (green).

Flights

Similar to vessels, airplanes can be tracked in flight and on the ground through surveillance technology. The data we gather regarding flights originates from the system named ADS-B, as described by [152]. We collect daily statistics on multiple airports throughout the world, using two sources.

Our primary source is the AeroDataBox API, which is available through [4]. This API collects data from external public data sources, community-maintained and commercial databases. Their collected data can be queried through an API made available on the RapidAPI platform.

Our secondary source is FlightRadar24, which can be accessed through [40]. This company describes itself as a global flight tracking service, collecting real-time data on thousands of air crafts.

We do not track all flights in the world, however. We focus on departures from commercial, cargo, and private flights. By focusing on departures, we attribute each flight to the airport, and hereby country and continent from which it departs. Also, we attribute the date and time to the departure time. This prevents counting flights twice and, as the plane will generally depart from its arrival airport at a later time, has a minimal impact on misclassifying flights to the correct country and timestamp. We exclude canceled flights and count code-shared flights as a single flight (in case one flight has multiple operators).

Regarding the tracked locations, we selected 24 major airports over the world. We did so by analyzing our collected data. An estimate of the flight volume per airport is also available through [150]. We selected a maximum of five airports per continent, a maximum of three airports per country, and a minimum of 20 million passenger volume in 2018. Section 4.6 lists all ICAO codes of the airports we track.

Figure 4.2 visualizes the flights data. First, we observe that the starting date of the graph lies around March. The first data point in our database is the 24^{th} of February. Second, we observe slight differences between the continents. Asia shows a relatively large number of flights since February, which can be explained by the fact

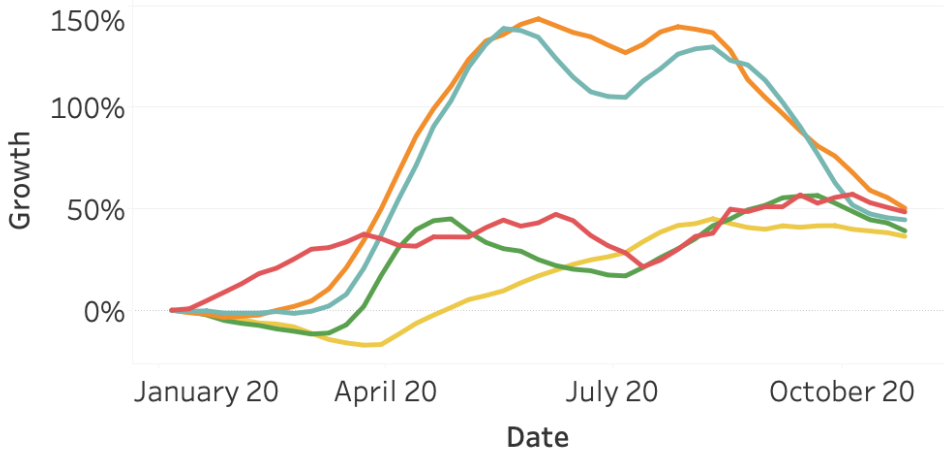


FIGURE 4.5: Online searches for bicycle through Google as of January 2020, split per continent. North America (orange), South America (yellow), Asia (red), Europe (light blue), and Oceania (green).

that the Covid-19 outbreak started earlier. We possibly miss the part of the initial decline. All other continents show a large decline in flight activity after the outbreak in March.

Vehicles

We utilized connected vehicle data from 57 countries, distributed over the different continents, with the majority of data collected in America, Asia, and Europe. The data is aggregated on a daily level, where we introduce the traffic intensity on a daily level as a representation of the number of active vehicles during the day. In other words, the activity during time window t is approximated by the average number of vehicles that operate during this time window. To reliably estimate the traffic intensity, we solely focus on cities where we selected 400 large cities distributed over the 57 countries.

Figure 4.3 visualizes the vehicle activity data. We observe a similar pattern to the flight activity. Most continents show a decline in vehicle activity in March, with the exception of Asia. After April, the traffic intensity of all continents is generally increasing.

Train and Bicycle Search Activity

Besides ships, airplanes, and vehicles, two remaining and major transport types are trains and bicycles. However, gathering data on these sources is challenging on a global scale.

Typically, usage of bicycles is hardly registered, as they rarely contain sensors registering their usage. The electrification in the bike industry might change this, but it is currently not available. A potential source could be fitness tracking apps. However, they typically focus on exercising and performance but hardly on commuting. Besides, they do not publish their data for research purposes.

Train usage is challenging to retrieve as this is highly dependent on the respective operator. Different countries typically have different operators, which typically have

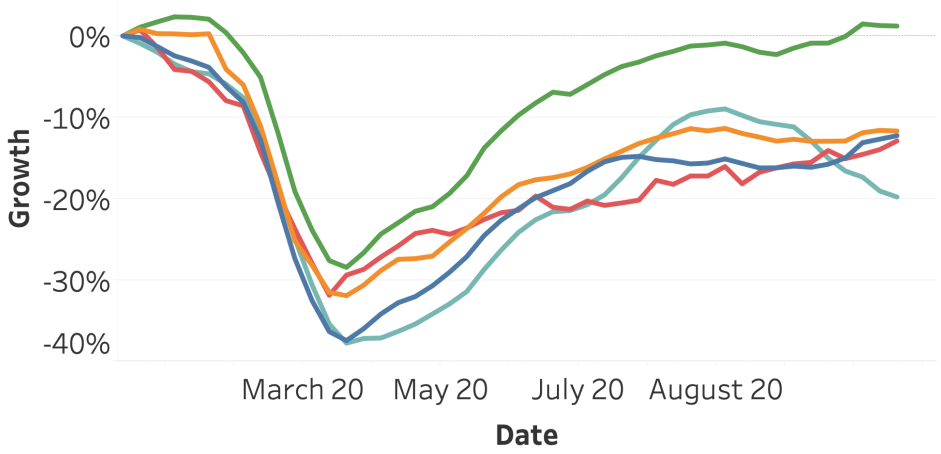


FIGURE 4.6: Average stock indices growth as of January 2020, split per continent. Africa (dark blue), America (orange), Asia (red), Europe (light blue), and Oceania (green).

a different method or granularity on which they publish data if they decide to publish these data. Given this research's global scope, we decided to find an alternative source to estimate train usage.

We estimate the usage of bicycles and trains by observing online search behavior. Google publishes this information on a global scale at [47]. Using this tool, we extract relative indices for online search behavior through Google on the topics bicycle and trains. We can do this over a time span of 2020, split by many countries in the world.

Figure 4.4 visualizes the Google search activity for the topic trains. Surprisingly, the impact of the Covid-19 outbreak on the search activity seems to be limited in South America, compared to the other continents. Additionally, in Europe, the activity grew to nearly the level of January 2020. However, it started decreasing again after August. This can be explained by the second peak of Covid-19.

Figure 4.5 visualizes the Google search activity for the topic bicycles. This seems to be the only mobility-related data source that has increased in activity after the first Covid-19 peak in 2020. For all continents, there is a positive growth relative to January. This growth is most notable in North America and Europe, which might partially be influenced by the seasonality. In summer, we expect more bicycle searches. However, both South America and Oceania show a rising search volume as of April 2020, despite the ending of summer. This growth might be explained by the decrease in public transport availability. Also, cycling is one of the few outdoor activities that are possible under most Covid-19 related measures. People that usually practice team sports like football, basketball, or field hockey might have switched to cycling.

Stock Markets

To obtain stock market information, we use the software implementation of [8]. For this, we tracked 40 major country indices, 74 raw materials, 566 stocks composed of the top-valued companies per index, and 148 currencies, of which 98 cryptocurrencies. The country indices are select from the Yahoo major world index list [153],

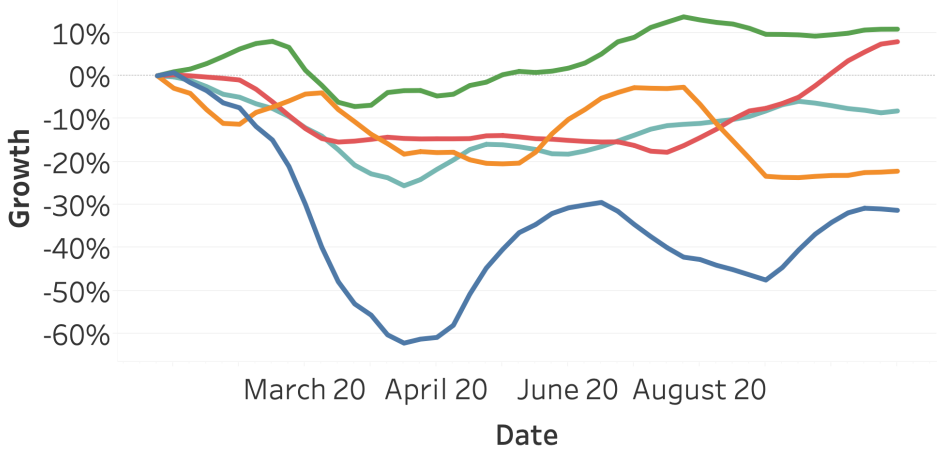


FIGURE 4.7: Average raw material growth as of January 2020, split per category. Energy (dark blue), Food & Fiber (orange), Grains (red), Livestock & Meets (light blue), and Metals (Green).

where we added the largest index of The Netherlands, Austria, Sweden, and Spain to increase the coverage in Europe. All stock information is tracked over a time span from *2020-01-01* to *2020-11-12*. We transformed all market close prices on a daily level from its listed currency to usd, using the close exchange rate between the foreign currency and usd.

Figure 4.6 visualizes the average stock index growth per continent. We can observe a joint decline for all continents, where Oceania reacts slightly slower and recovers faster compared to the other continents. After the 20^{th} of March, the stock indices on all continents are generally increasing.

Figure 4.7 presents the average stock index growth for various groups of raw materials. Apart from the energy sector, all material groups decline less extensively compared to the average index per continent (Figure 4.6). The energy category suffers from a strong decline of more than 50 percent. Metals experience the lowest decline and the earliest recovery.

Covid-19 Cases

One of the most central datasets of this research is the global Covid-19 dataset. Our primary source is the Covid-19 data repository of [32], sourced by the Center of Systems Science and Engineering (CSSE) at Johns Hopkins University. We extrapolated the absolute registered corona cases and related deaths per country on a daily level. We normalized the absolute deaths to deaths per 100,000 inhabitants to account for strong differences of inhabitants per country.

Figure 4.8 visualizes the normalized death per continent on a weekly level. Europe shows a strong first peak around the end of April and a second one at the start of November. Surprisingly, Europe shows a steep increase, followed by a strong decline in the number of deaths. Contrary to Europe, North America shows more consistent growth in the number of deaths and less intense waves. Surprisingly, South America, Asia, and Oceania observe far fewer deaths. This might partially be explained by differences in measuring and registering Covid-19 deaths.

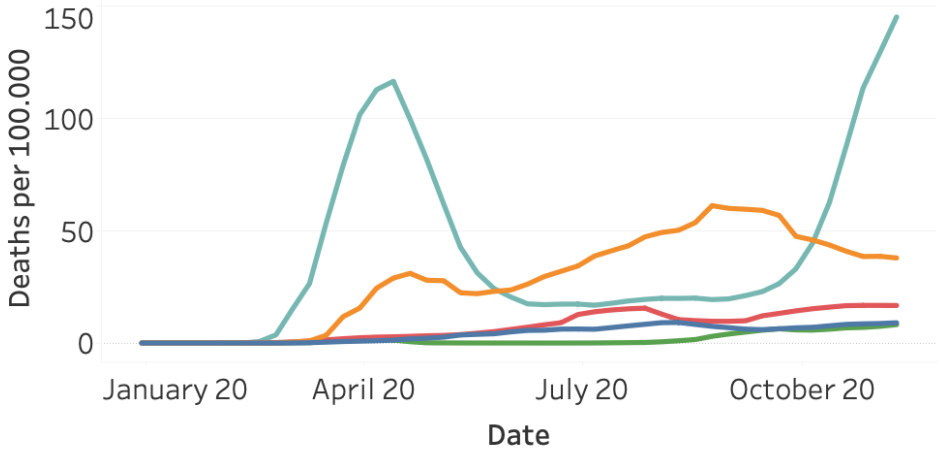


FIGURE 4.8: Covid-19 deaths per 100,000 inhabitants as of January 2020, split per continent. North America (orange), South America (yellow), Asia (red), Europe (light blue), and Oceania (green).

Covid-19 Measures

The measures to fight Covid-19 are strongly varying over time and between different countries or regions. To the best of our knowledge, there is no universal dataset available with all corona measures on a country level. Therefore, we limit the data collection of Covid-19 measures to The Netherlands in isolation. For this, we mainly extrapolated the measures from the different press conferences, using the website of the Dutch National Institute for Health and the Environment (RIVM) [102].

From this, we obtained the following features: press conference date, the date the measures take effect, the opening or closure of the primary schools, the secondary schools, the universities, indoor sports, outdoor sports, and professions with close contact such as hairdressers. In addition, we construct features for the number of allowed visitors in restaurants, churches, home settings, and public spaces such as concert halls and theaters.

Figure 4.9 visualizes the opening (green) and mandatory closure (blue) for different segments.

4.3 Methodology

To answer our research questions, we split our methodology into three parts. First, we combine the various mobility-related data sources into one dataset. Second, we add corona and stock measures and investigate various relations in the resulting dataset. Third, we investigate whether we can quantify the impact of the Covid-19 measures taken in the Netherlands.

4.3.1 Combining Mobilities

Combining the mobility-related data sources is a challenge, as their origin and data structures are not aligned. We tackle this by defining a base dataset to which all sources can be mapped. This dataset consists of the dimensions date and country.

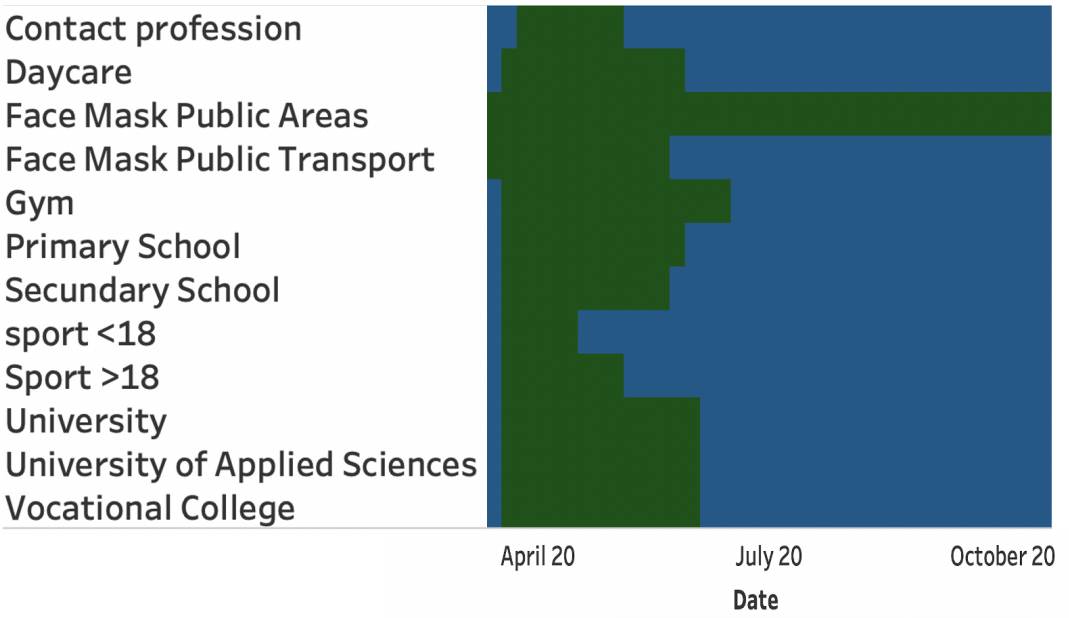


FIGURE 4.9: Opening and mandatory closure for different business segments. Opening (green), and mandatory closure (blue).

We apply filters corresponding to the most limited dataset, which in our case concerns flights. This dataset contains dates starting from the end of February, and is limited to a selected number of airports and hereby countries. However, we still cover major countries and the largest part of 2020. We aggregate the various sources to a daily and country level, average the corresponding measure, and merge the result to the base dataset. The resulting dataset will contain for each country and date the vessel, flight, vehicle, bike, and train measures. To adjust for weekly seasonality, we aggregate this set to a weekly level.

4.3.2 Relation Between Variables

One of the goals of this chapter is to find relationships between all statistics gathered in this research. We do this by combining the data in an appropriate table, preprocessing its contents, and applying a correlation test and dynamic time warping to it.

First, we expand the mobility dataset we created in Subsection 4.3.1 with corona cases, corona deaths, and stock indices. We filter the stock data only to contain country-related indices and average all indices within one country.

Second, we preprocess the dataset. We standardize the measures to having a mean of 0 and a standard deviation of 1. In addition, we compute lag variables. For each measure, we compute *measure_lag_i* with $i \in \{1, 2, \ldots, 5\}$, in which the corresponding measure is lagged by i weeks.

Finally, we investigate the relation between the resulting measures in terms of Pearson correlation and dynamic time warping. As described in [33], dynamic time warping is a robust distance metric that allows an elastic shifting of the time axis,

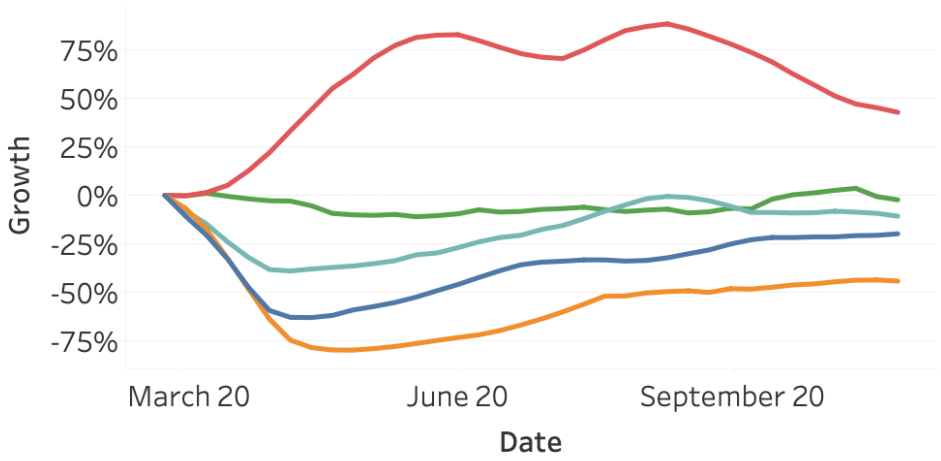


FIGURE 4.10: Growth in global usage of mobility relative to March 2020, split per mobility type. Bicycle (red), vessel (green), train (light blue), traffic (blue), and flights (orange).

to accommodate sequences that are similar. It compares points between two sequences in a many-to-one fashion, in contrast to the one-to-one fashion of the Pearson correlation. We implemented this procedure through the Python implementation of [44].

4.3.3 Impact of Covid-19 Measures

Quantifying the exact impact of Covid-19 measures is challenging as we solely have the Dutch measures available, and many other causal effects play a role. We aim to provide insights into the relation between corona deaths, mobility usage, and corona measures. The constructed dataset contains the maximum allowed visitors for specific sectors and presents which sectors are closed or open. Due to the limited sample size, a statistical test will not provide reliable and conclusive results. Therefore, we aggregate all data on a weekly level and visualize the major changes in corona measures within the chart. With this, we aim to present potential relationships between the three variables.

4.4 Results

In this section, we highlight our results in the same structure as described in the methodology.

4.4.1 Combining Mobilities

Figure 4.10 highlights the growth in the measured usage of the monitored mobility types, relative to the first of March 2020. The usage is measured in the manner described before and averaged on a global scale. We can observe the usage of all mobility types, except bicycles, has declined since March 2020. Flights show the largest decline, followed by traffic and train. Vessel activity has not decreased much – all in large contrast to the usage of bicycles, which shows a large increase in usage.

TABLE 4.1: Original time-series (lag=0) on rows and lagged variable on columns. The diagonal, as well as correlations that are not significant ($p > 0.05$) are omitted from the table. We present each correlation along with the best lag for the column variable (round brackets) and its significance level: a ($p < 0.05$), b ($p < 0.01$), c ($p < 0.001$) (superscript).

	Traffic	Cases	Deaths	Flights	Bicycles	Train	Stock	Vessel
Traffic		0.26^{a} (5)	-0.71^{c} (0)	**0.86**^{c} (0)	-0.70^{c} (0)	**0.86**^{c} (1)	0.82^{c} (3)	0.34^{b} (0)
Cases			**0.58**^{c} (0)		0.32^{b} (5)	-0.35^{b} (2)		
Deaths	-0.77^{c} (1)	0.58^{c} (0)		-0.74^{c} (0)	0.73^{c} (0)	**-0.85**^{c} (3)	-0.74^{c} (2)	
Flights	**0.94**^{c} (1)		-0.74^{c} (0)		-0.78^{c} (0)	0.91^{c} (3)	0.84^{c} (4)	0.39^{b} (0)
Bicycles	-0.85^{c} (3)		0.81^{c} (2)	**-0.86**^{c} (2)		-0.85^{c} (5)	-0.85^{c} (5)	-0.42^{c} (0)
Train	0.75^{c} (0)	-0.32^{b} (0)	-0.70^{c} (0)	0.54^{c} (0)	-0.56^{c} (0)		**0.84**^{c} (0)	
Stock	0.73^{c} (0)	0.34^{b} (5)	-0.59^{c} (0)	0.34^{b} (0)	-0.46^{c} (0)	**0.84**^{c} (0)		-0.33^{b} (4)
Vessel	**0.47**^{c} (2)	0.26^{a} (5)	-0.33^{b} (3)	0.45^{c} (3)	-0.48^{c} (1)	0.29^{b} (4)	0.35^{b} (5)	

4.4.2 Relation Between Variables

Table 4.1 presents the strongest Pearson correlation coefficient, between the original variable on the rows and the lagged variable on the columns. All variables on the diagonal, as well as any non-significant ($p > 0.05$) variables, are omitted from the table. We can observe some strong correlations across all variables where most are significant. We see, for example, that both trains with lag 3 (columns) and traffic with lag 1 (columns) highly correlate with flights (rows).

More interestingly, we can observe that corona deaths seem to correlate more with the other variables, compared with corona cases. In addition, corona-related deaths seem to have a direct negative correlation with traffic, flights, trains, and stocks and a lagged negative correlation with vessels. The lagged deaths have a positive correlation with bicycle searches.

Figure 4.11 highlights the correlation of traffic, flights, and stocks with the lagged corona deaths. The lag of the corona death is presented on the x-axis. In other words, it presents the correlation of the target variable with the corona deaths of x weeks earlier. We can observe strong negative correlations of the target variables with the lagged corona deaths. All three variables show a similar trend where the negative correlation decreases as the lag of corona deaths increases. This indicates that, for example, the reduced traffic intensity due to an increase in corona death is almost recovered to the initial level after five weeks. In a similar way, flight intensity recovers, but perceives a less strong recovery. This could potentially be due to the fact that flight are booked and scheduled in advance, resulting in an increased recovery time compared to road traffic. Moreover, flights potentially experience more corona restrictions as they often cross borders. Stock prices recover faster compared with traffic and flights.

Figure 4.12 shows the dynamic warping alignment plot of vehicle activity and corona deaths in the Netherlands. This indicates that the corona death increase is running ahead of the vehicle activity decline. It presents that the vehicle activity reacts slowly to the number of corona deaths.

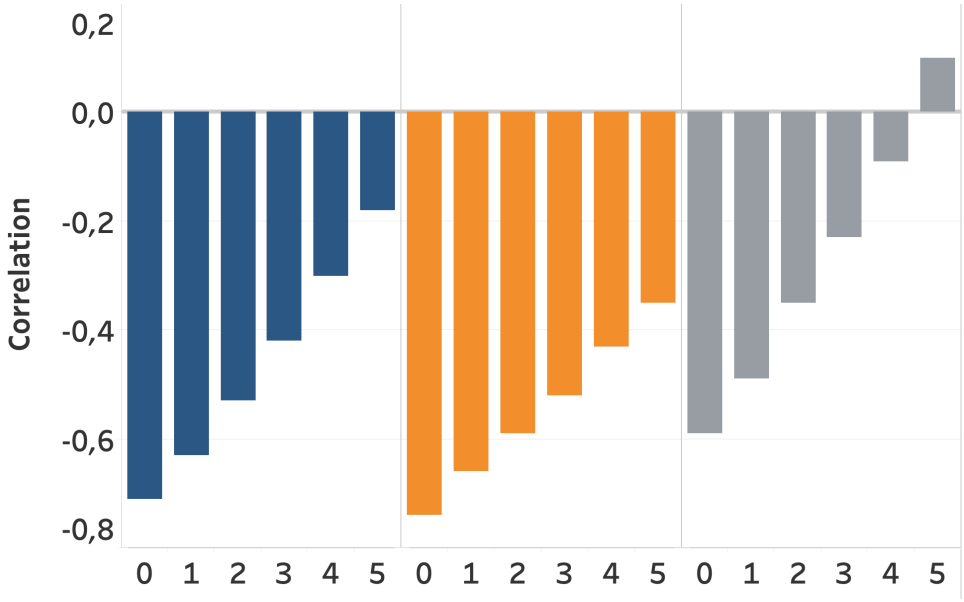


FIGURE 4.11: Pearson correlation coefficient of traffic, flights, and stocks, with the lagged corona deaths variable on the x-axis. It presents the correlation between the target variable and the corona deaths of x weeks earlier. Traffic (blue), Flights (orange), and Stocks (grey).

4.4.3 Impact of Covid-19 Measures

Figure 4.13 highlights the corona deaths (top view) and the vessel, traffic, flight activity (bottom view), in relation to the largest changes in corona measures. A steep decline in both traffic and flights is visible after the first measures (closing all education, public areas, and sports) take effect. The decline in mobility continues as the maximum group sizes are further reduced to three people, and close contact professions are closed. Slightly after the first peek in deaths, a series of measures is reduced, resulting in a steady increase of mobility (3-5). A stable summer with a low number of death and a slow increase in traffic, follows after removing the maximum number of visitors for restaurants and reopening the gyms (6).

Surprisingly, the second wave of corona death with less strict measures that solely reduce the number of visitors and gradually maximizes the allowed group sizes from six to two (12-14), did not result in a decline in mobility. A rather stable pattern can be identified, deviating from the first wave.

4.5 Discussion

In this chapter, we presented and analyzed various relations between Covid-19, mobility, and stock-related data sources. Collecting and comparing such a wide range of data can only be done by making compromises. In this section, we will discuss our results and main design choices.

Overall, the Pearson correlations in Table 4.1 show significant relations between most variables. Especially Covid-19 deaths show strong correlations. This is in contrast with the Covid-19 cases, which show fewer significant and overall weaker

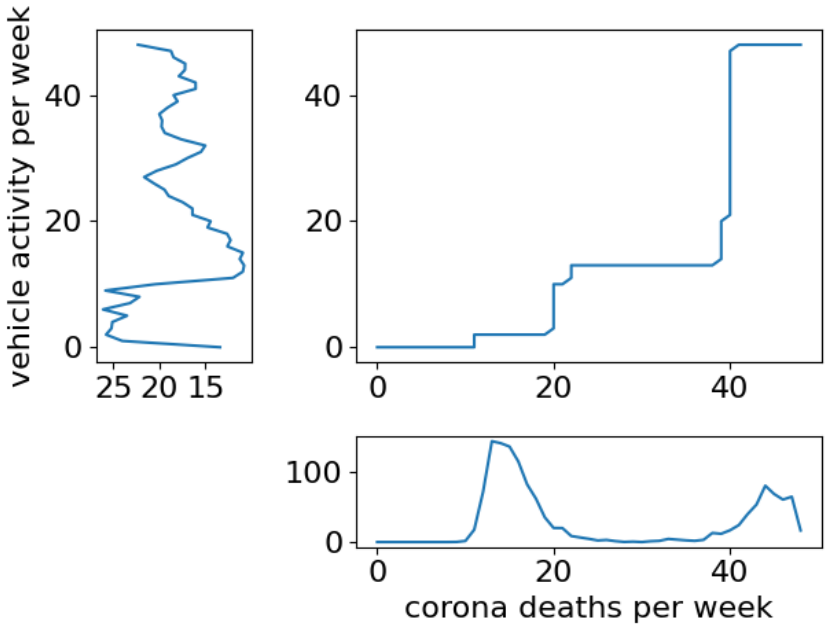


FIGURE 4.12: Dynamic warping alignment plot between vehicle activity per week and corona deaths per week in the Netherlands.

correlations. This might be explained by deaths being registered more accurately or cases having a higher dependency on testing strategies.

A strong negative correlation between Covid-19 deaths and mobility with a low lag indicates that more deaths rapidly influences mobility in a negative way. The only exception is bicycle searches with has a strong positive correlation with a small lag of two weeks, indicating that bicycle searches do not rapidly spike after the corona deaths increase. Furthermore, we presented that traffic, flights and stock are negatively influenced by increased corona deaths. But that this correlation decreases linearly per week, where traffic is almost recovered to its original level within five weeks.

We present a strong correlation between death and the number of flights. However, we have to state that this possible causal relationship is difficult to measure due to the corona measures influencing this relation. Our reporting contains aggregations over multiple countries, being directly impacted by the measures taken in all countries.

Deaths seem to be a more reliable estimator compared with cases. This could potentially be related to different approaches for the registration of corona cases. However, the number of people in intensive care with corona might be a more reliable metrics compared with the number of cases and the number of death. Unfortunately, to the best of our knowledge, there is no publicly available dataset for the number of intensive care patients.

Regarding the flight data, we only have statistics available from February 2020 onwards. Ideally, we would cover the complete year, also as our other data sources did start in January 2020. However, our data sources do not allow us to go further back in time than six months. We could consider adding a ternary source for the flight data. However, this adds complexity and additional cost to the project.

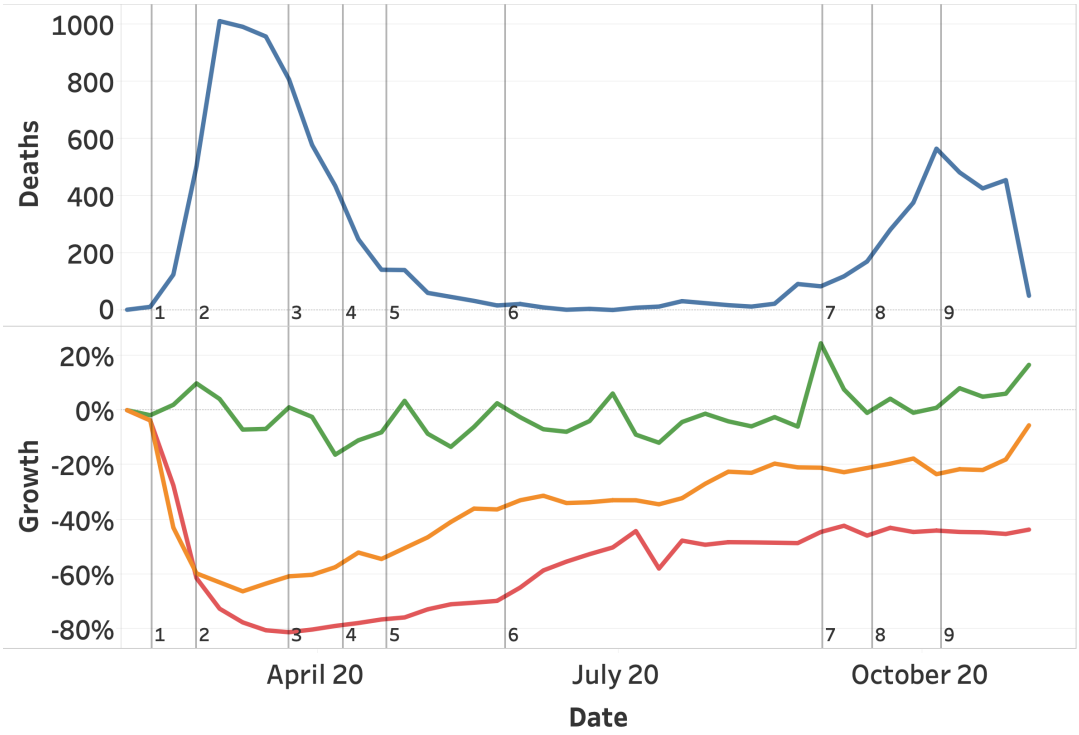


FIGURE 4.13: Corona deaths with growth in vessel, traffic, and flights with respect to the major changes in corona measures. Corona deaths (blue) and growth in vessel (green), traffic (orange), and flights (red). (1) closing all education, sport, and public areas. (2) reducing group sizes from 100 to 3 and closing close contact professions. (3) Opening sports under 18 years. (4) Opening close contact professions and sports for adults. (5) Open restaurants, increase group sizes from 3 to 6 and introduce mandatory face masks in public transport. (6) remove maximum visitors for restaurants and open gyms. (7) Reduce restaurant capacity to 30. (8) Close restaurants and max group sizes to 4. (9) Max group sizes to 2.

Regarding the stock data, it has to be noted that stock markets do not operate in isolation. Markets in different time zones react to each other, resulting in some causal effects which are not identifiable in our data. We overcome this to a large extent by aggregation on a weekly level, but a small unmeasured causal effect remains.

In our current analysis, we do not take into account yearly seasonality. This seasonality would be relevant in making a more accurate comparison between the usage of the various modalities in 2020. For example, we know that bicycle usage correlates with the weather, showing an increase in summer usage each year. However, we did not include the seasonality aspect in our research as the majority of our data sources have a time window of less than a year. Including seasonal relationships would require a time window of at least two years to reliably estimate the impact. Therefore, a year from now, further research could focus on extending the dataset horizon to analyse seasonal patterns in the relationships.

In our current analysis, we do not include detailed statistics on a country or daily level. Ideally, we would have covered this analysis in more detail, but we decided to keep statistics compact and high-over. Exploring all variables on a lower level of detail would reduce the readability of this chapter. Therefore, we would suggest

further research based on the initial findings. Further research could focus on exploring the presented relation between death and flights on a country level, to better estimate the causal relationship. Furthermore, we presented strong relationships for Covid-19 deaths, in contrast with Covid-19 cases. Further research could could focus on the registration of Covid-19 cases and testing strategies, to evaluate how this weaker relationship is established.

4.6 Appendix: Tracked Airports

List of tracked airports by ICAO code: ZBAA, OMDB, RJTT, ZSPD, ZGGG, EGLL, LFPG, EHAM, EDDF, LEMD, KATL, KLAX, KORD, CYYZ, MMMX, SBGR, SKBO, SPJC, SCEL, SBSP, YSSY, YMML, YBBN, and NZAA.

These correspond to the cities of Beijing, Dubai, Tokyo, Shanghai, Guangzhou, London, Paris, Amsterdam, Frankfurt, Madrid, Atlanta, Los Angeles, Chicago, Toronto, Mexico City, Sao Paulo, Bogota, Callao, Santiago, Sydney, Melbourne, Brisbane, and Auckland.

5 Predicting Travel Behavior by Analyzing Mobility Transactions

5.1 Summary

Urban planning can benefit tremendously from a better understanding of *where*, *when*, *why*, and *how* people travel. Through advances in technology, detailed data on the travel behavior of individuals has become available. This data can be leveraged to understand why one prefers one mode of transportation over another one. In this chapter, we analyze a unique dataset through which we can address this question. We show that the travel behavior in our dataset is highly predictable, with an accuracy of 97%. The main predictors are reachability features, more so than specific travel times. Moreover, the travel type (commute or personal) has a considerable influence on travel mode choice.

5.2 Introduction

The analysis of mobility is of key importance to tackle major urban planning challenges [**sb_ant_paper**]. It is projected that by 2030, today's 1.2 billion global car fleet could double [16]. This has a major impact on the dynamics in urban areas: traffic delays, unhealthy smog levels, noise, routine irritations of urban lives, and others. The introduction of other modes of transportation can help to alleviate these mobility challenges. One can think of public transport, bikes, and trains, as well as shared services or combinations thereof. In addition, [23] introduces many other (sub)urban transformations and transit-oriented developments to improve urban lives. At the same time, it is argued that in order to decide where to invest in requires a good understanding of the travel behavior of individuals and their underlying reasons.

Much research has been devoted to mobility patterns within cities. The gravity model is the prevailing framework for discovering and modeling these patterns [155]. This model is rather data-intense, in the sense that it requires specific parameters fitted from a continuous collection of traffic data. When these measurements are not available or not complete, the gravity model cannot be applied. In response to that, trip distribution models have been introduced [31, 136]. However, these models also rely on context-specific parameters. In [144], the authors explain how radiation models [121] can model mobility patterns based on only the population's spatial distribution as input.

Through advances in technology, trip information per individual can be recorded more precisely. E.g., telematics modules in cars can record the time and location when a car starts or turns off the engine. Travel cards issued by companies store information on the usage of public transport, bikes at the start of the trip, and the

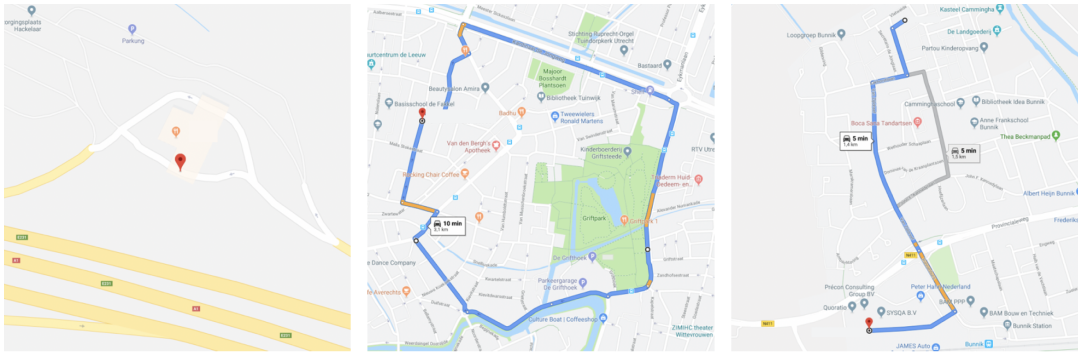


FIGURE 5.1: Three categories of trips that are filtered from the data: (a) gas stations; (b) round trips; (c) relatively short trips.

end of the trip. Such detailed information per individual allows one to perform mobility analysis on a much more personalized level. The previously mentioned mobility models cannot handle such a granularity of information and cannot be applied. Models have been introduced to analyze trip data from one mode of transport to reveal travel patterns [83, 46]. However, such analyses do not reveal the underlying reasons why an individual chooses one mode of transport over another one. In the literature, there is a gap in analyzing these mobility choices at such granularity due to the lack of high-quality data.

In this chapter, we analyze the travel mode choices based on a unique dataset consisting of mobility transactions on an individual level. This allows us to follow individuals throughout the day and the year. We propose a measure to quantify the speed of any transport type for any neighborhood and show how to combine relevant external data sources. By doing so, we can accurately predict travel mode choices. Our model shows the opportunity to influence travel mode choices and gives the potential to simulate the impact of infrastructure changes.

This chapter is organized as follows. Section 5.3 describes the dataset and the data preparation. Section 5.4 illustrates the impact of the data preparation and introduces the models for understanding mobility. The results of the analysis are discussed in Section 5.5. In Section 5.6, conclusions and recommendations for further research are presented.

5.3 Data

The data used in this research is gathered from multiple sources by a company that provides mobility to customers through a mobility card. Individuals can use different travel modes using this card. The card enables one to use a car, all forms of public transport, a taxi, car sharing, and bike sharing. The customers using this service are all employed by one specific company in the Netherlands, which has multiple offices spread throughout the country. Travel usage is registered through automated systems and stored as transactions.

Car transactions are registered automatically by a built-in telematics module in the cars, which have trip registration as their sole purpose. All collected transactions are considered private; however, under strict conditions, analysis of this data is allowed. Consequently, due to these conditions, we cannot directly determine the identities behind the person identifiers in the data.

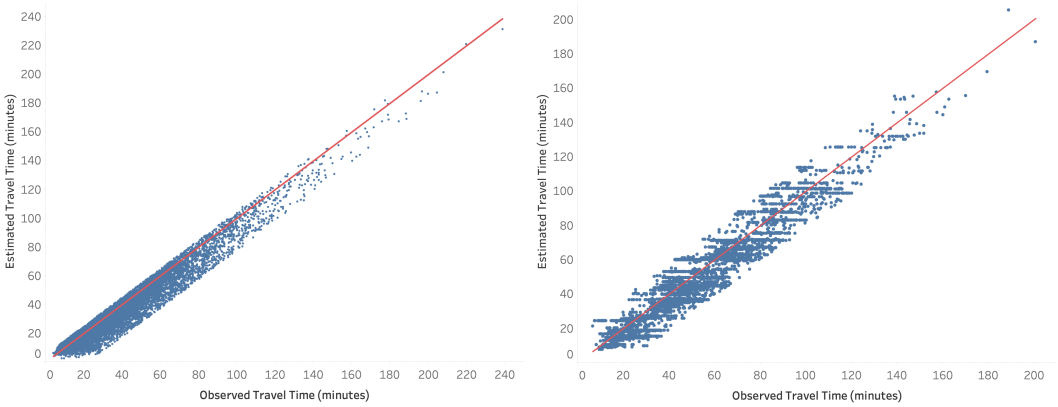


FIGURE 5.2: Observed vs estimated travel time: (a) car; (b) public transport.

The full dataset contains over half a million mobility transactions from over a thousand employees. This concerns a period of one entire year 2018. We filter the data by individuals having access to both public transport and a telematics-enabled car. For each mobility transaction, we know the transport type, start and end date and time, start and end location, and costs. We have aggregated statistics for each individual, such as the city of residence, lease category, and commute mileage. Other individual-specific attributes such as age, gender, and fuel compensation are not taken into account for privacy reasons.

Analyzing such a dataset imposes various challenges. In the next section, we discuss the data cleaning. Then we explain how to estimate statistics on the alternative travel mode. Repeating choices will be discussed in the subsequent section, followed by a method to compute the start and end locations of public transport transactions more accurately. We conclude with an examination of relevant external data sources.

5.3.1 Data Cleaning

The dataset consists of all transactions of all individuals for the year 2018. However, we do not consider all transactions in this research for various reasons. Specifically, three categories of transactions are filtered: transactions to gas stations, transactions with a similar start and end location, and relatively short transactions. Figure 5.1 visualizes the three categories. We filter out these transactions for reasons that we will explain next.

Transactions departing or ending at gas stations are filtered as they are not considered the 'true' start or destination of a transaction. Typically, it is a compulsory stop en route to a different destination. As all locations of gas stations in the Netherlands are publicly available, these transactions can be filtered easily.

Transactions with a similar start and end location (within one transaction) are difficult to analyze as we do not know what happened during the trip. It is impossible to calculate accurate statistics on alternative travel modes. Therefore, we filter transactions of which the start and end location are within a distance of 200 meters.

Relatively short transactions are not considered either, since the availability of alternative modes can be questioned. Additionally, our focus is not on these short trips.

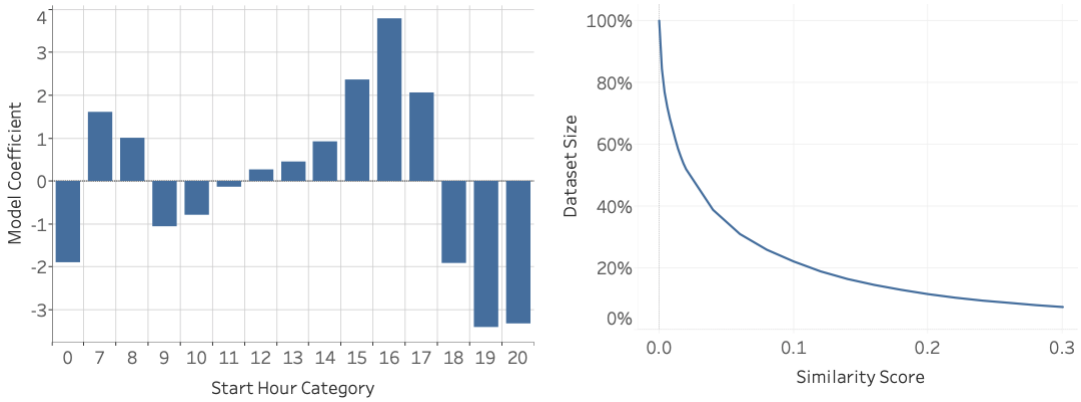


FIGURE 5.3: Data processing: (a) model coefficients for re-estimating car trips; (b) trade-off between variety and availability of data.

For example, if a transaction by car exists with a length of one kilometer, we could compute statistics on public transport on this transaction. However, the resulting statistics could be similar to those of walking. Therefore, we decided to filter all trips with a distance shorter than four kilometers.

5.3.2 Estimating Statistics on the Alternative

The transactions contained within the dataset show statistics on the chosen mobility type. For scenario analysis, we are interested in the statistics on the alternative. Specifically, we are interested in the travel time, distance, and CO2 emissions. We compute these by using external APIs. A wide variety of them is available, including the Open Routing Service, Google Maps, Bing Maps, City Mapper, and Tripgo. We chose to use the HERE API [56] for estimating statistics on car alternatives and the TravelTime Platform API [143] for public transport. This choice is made based on the availability and cost of the services. An estimate of CO2 emissions is made for cars by analyzing all historical transactions of the individuals (including liters tanked), and for public transport by using statistics from research on emission factors in the Netherlands [81].

The performance of these APIs can be measured by requesting statistics on known transactions and comparing those to the observed statistics. Figure 5.2 shows a comparison of observed and estimated travel times for (a) cars and (b) public transport. The red line is used as a reference and has slope 1 in both graphs. Interestingly, estimated car travel times are slightly underestimated. This can be partially explained by congestion. Concerning public transport, we can see that there is a significant variance in the observed travel time for similar estimates. This can be explained by irregularities in schedules and varying arrival times of individuals at stations.

To improve the accuracy of the estimates, we create a linear model on top of the API estimation. For cars, this model is based on the API estimation, start hour of the transaction, and a weekend indicator. For public transport, this model is based on the API estimate and start hour of the transaction. The choice for these features is based upon the significance of their results. The linear model results in an increase of the coefficient of determination R^2 from 0.835 to 0.873 for cars and from 0.759 to 0.814 for public transport.

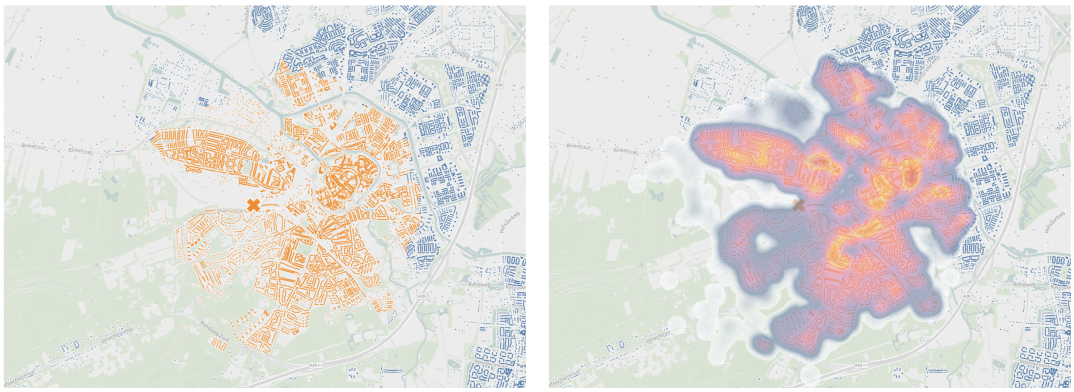


FIGURE 5.4: Estimating start and end locations: (a) addresses inside (orange) and outside (blue) ten minutes cycling from a station (orange cross); (b) sample density of households within ten minutes cycling range.

Figure 5.3 (a) shows the model coefficients for re-estimating car trips. Interestingly, there is a clear relationship between the start hour of a trip and the fitted model coefficient on the data. During rush hours, in the morning and in the afternoon, the model coefficients are positive; outside rush hours the coefficients are negative. The highest model coefficient corresponds to the hour that is often considered as the hour with the highest congestion level: 16:00. To simplify the model, we chose to group all hours during the night in a category labeled as '0'.

5.3.3 Start and End Locations

The start and end location of the transactions are difficult to interpret, as they only provide an estimate of the 'true' start and end location. For cars, these locations are generally close by, as parking spots are widely available. However, for public transport, this can be assumed to be less accurate as the transactions provide us with the check-in and check-out locations. These locations are always at stations.

To improve this estimation, we re-estimate the locations of public transport transactions. Using the TravelTime Platforms Time Map feature, we calculate the area that can be reached from each station in the Netherlands by ten minutes of cycling. Combining this with a data source containing coordinates of all addresses in the Netherlands (BAG), we compute all reachable addresses for all stations in the Netherlands. Next, we sample one address for each transaction concerning public transport from all reachable addresses from the corresponding station. We use this address instead of the address that is shown in the raw data.

This process is visualized in Figure 5.4. On the left (a), all addresses within ten minutes of cycling from a station are visualized. The station is located at the orange cross, all addresses reachable within the ten minute threshold are colored orange and the others blue. The orange area resembles a circle, however, this is not necessarily true. In some areas, there might be natural obstacles or little infrastructure. This will influence the travel time towards these areas, and hereby the shape of the area. On the right (b), the sample density is shown. In some neighborhoods, the addresses are more densely packed and, therefore, should have a higher probability of being selected. If we sample at random from all known addresses, we automatically correct for the population density.

TABLE 5.1: Quantifying repeating choices.

Time PT	Cost PT	CO2 PT	Time car	Cost car	CO2 car	Similarity
77.42	22.3	7.82	42.21	4.90	7.83	n/a
77.36	22.3	7.82	42.29	4.91	7.85	0.001
77.78	22.3	7.82	42.24	4.91	7.84	0.002
78.59	22.3	7.82	42.10	4.90	7.83	0.006
42.26	9.61	3.37	28.56	2.97	4.74	0.788
92.86	15.3	0.41	63.37	8.75	14.0	0.922
138.6	32.6	0.88	109.3	19.8	31.7	2.036
198.0	49.0	7.77	120.8	23.1	36.8	2.421

5.3.4 Repeating Choices

A potential challenge with fitting models on the transactions is that the model becomes biased towards choices that are often repeated. For example, a person might decide once on his or her commuting transport mode and hereafter execute it hundreds of times. In contrast, an occasional trip to a specific destination might only appear once. We want to present a dataset with variety to the models in order to prevent the models from being biased towards choices that are repeated often. We cannot filter based on the trip type, as individuals can commute to multiple offices, can have business trips to similar locations, or change behavior over time.

To increase the variety of the data, we remove rows that are highly similar to others. For this, we define the distance $d(r_i, r_j)$ between record r_i and r_j as in Equation 5.1:

$$d(r_i, r_j) = \frac{f * |r_i - r_j|}{s}. \tag{5.1}$$

In this equation, f is the vector with the feature importance acting as a weight, and s is the feature standard deviations. The vector f is based on the features of the model developed in Experiment 2 in Section 5.4.2. We sort the data by date and time, partition by person id, and remove all rows that have a distance lower than a certain threshold, for index $j > i$.

Determining the threshold implies balancing the variety and availability of the data. Having a dataset with a large variety implies having low volume; having a dataset with high volume implies having little variety. Figure 5.3 (b) visualized this trade-off by showing the relation between the threshold and dataset size. A high similarity score as threshold implies more choices are seen as similar, which results in lower data volume. We empirically set the threshold to 0.04, which keeps roughly 30% of the data.

Table 5.1 shows an example of how the repeating choices are identified. Taking the first row as a reference, the distance between the following rows is computed according to Equation 5.1. Taking the threshold of 0.04, rows 2, 3, and 4 will be filtered. This does not guarantee the other rows will not be filtered, as the process will be repeated from row 5 onward.

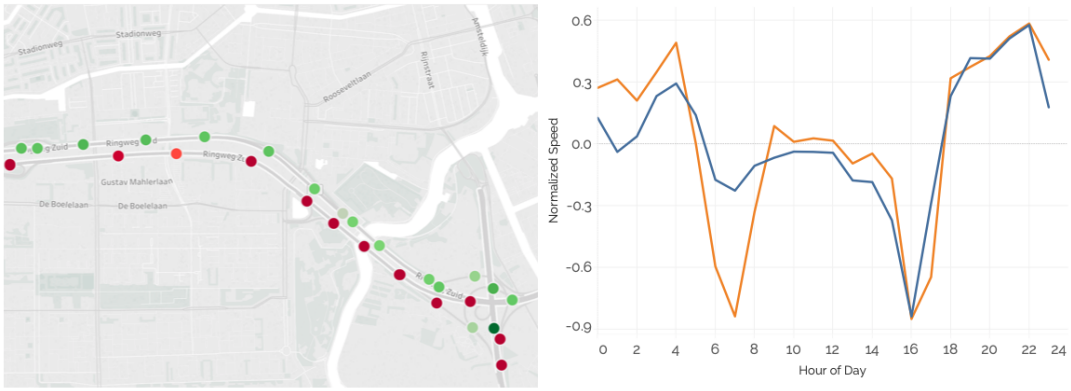


FIGURE 5.5: Measuring congestion: (a) measurement locations: having high (red) and low (green) congestion levels; (b) congestion levels: from Amsterdam to Almere (blue) vs from Almere to Amsterdam (orange).

5.3.5 External Sources

Besides the transactions and personal statistics, we use external data sources to calculate features. This concerns data on congestion, reachability of neighborhoods, and weather conditions.

Congestion is measured using data from the Dutch Nationale Databank Wegverkeersgegevens (NDW). The NDW continuously measures the speed and volume of cars driving over their sensors on federal roads. This concerns 37 thousand sensors across the Netherlands, which report statistics by the minute. Figure 5.5 visualizes this data. On the left (a), it shows the measurement locations on a highway around Amsterdam. The congestion level is indicated by color, red meaning high congestion and green meaning low congestion levels. Clearly, the highway is congested in a single direction. On the right (b), it shows the normalized speed on all measurement sites between two Dutch cities (Amsterdam and Almere) in opposite directions. The graph shows that congestion in the morning is heavy in one direction, whereas the opposite direction is hardly congested. This data allows us to quantify congestion on the road at a specific time, between the start and end locations, and in the corresponding direction for all transactions in our dataset.

A second feature we compute is the so-called *reachability* of neighborhoods. This captures the general speed of a particular transport type in a certain area. To compute these, we start by taking the definition of a neighborhood from the Dutch CBS. These neighborhoods are similar to postal code definitions. However, it provides a higher detail level and is still feasible for this analysis. For each neighborhood in the Netherlands, we compute the speed (distance over time) at which the 4,000 surrounding neighborhoods can be reached by both a car and public transport. To compute the travel time, we use the APIs selected in Section 5.3.2. To compute distance, we take the celestial distance. Both measures are calculated between the building lying closest to the center point of the neighborhoods. Next, we average these speed values to gain one numeric value per neighborhood. The resulting measure is visualized in Figure 5.6. It shows the reachability of the neighborhoods in the Randstad region in the Netherlands by (a) public transport and (b) car.

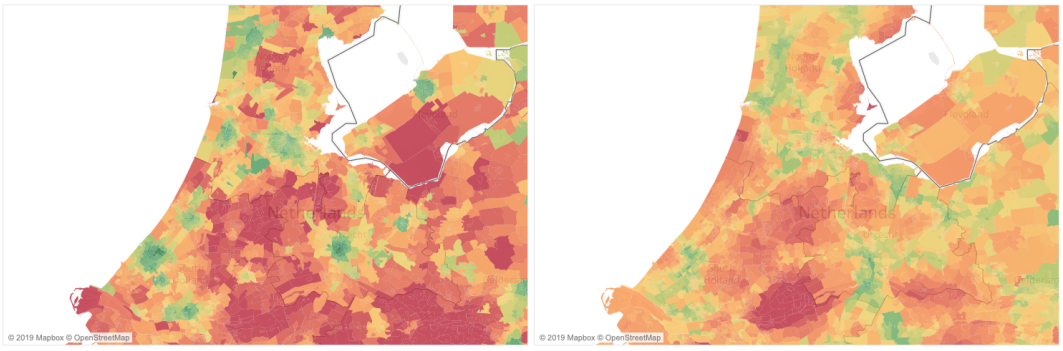


FIGURE 5.6: Measuring reachability of neighborhoods: (a) public transport; (b) car.

Finally, we add statistics on weather conditions. This includes wind, rain, temperature, sunshine, wind speed, and rain duration. These statistics are historically made available by the Dutch KNMI. They are measured on 50 locations spread throughout the Netherlands. For each transaction, we take the measurement values from the nearest station to the middle coordinate of the transaction.

5.4 Numerical Experiments

The data processing steps are pre-requisites to understand better and to predict travel behavior. Therefore, we create multiple models for describing the mobility transactions, define multiple experiments to assess the impact, and evaluate them accordingly. These three steps are described in this section.

5.4.1 Models

We fit five different models on the mobility transactions. The first model is a model familiar to the transport science field, a logit choice model. The other four models are commonly used in the machine learning field: logistic regression, feedforward neural network, gradient boosted decision trees, and random forest. The alternatives for all models are using the car or using public transport, making it a binary problem. The models are evaluated using five-fold cross-validation and are implemented in Python. The logit choice model using the PandasBiogeme [14] package, the logistic regression, neural network, and random forest using the scikit-learn [97] package, and the gradient boosted trees using the xgboost [26] package. The logit model is highly similar to the logistic regression model, however, they are implemented through different libraries. The model parameters are determined by a grid search procedure, the feedforward neural network performs best with a single hidden layer containing ten neurons.

5.4.2 Experiments

To highlight the impact of the data processing, we define four experiments. We start by fitting models on relatively raw data and step-by-step work through the processing steps to highlight their impact. The final experiment can be considered the most realistic and important.

- In Experiment 1, we take the raw data, filter it (Section 5.3.1), calculate features on the alternative (Section 5.3.2), and fit the models. The features used by the models are the travel time, costs, and CO2 emissions of both alternatives (car and public transport).
- In Experiment 2, we take the data from Experiment 1, change the start and end locations of public transport trips (Section 5.3.3), re-calculate the features on the alternatives, and re-fit the models of Experiment 1.
- In Experiment 3, we take the data from Experiment 2, filter by removing repeating choices (Section 5.3.4), and re-fit the models of Experiment 2.
- In Experiment 4, we take the data from Experiment 3, add features, remove correlated features, and re-define the models of the previous experiments. Added features are as described in Section 5.3.5, combined with the aggregated personal statistics, and a classification of the transaction as indicated by the individuals (private, commute, or business).

5.4.3 Evaluation

For each experiment, we fit the models on the data. We evaluate the performance by measuring the accuracy and the AUC [51]. Additionally, we use SHAP [86] to measure the impact of all features for the machine learning models. The SHAP-value represents the impact on the model output. For each feature, it holds that the larger its absolute SHAP-value, the larger its importance.

5.5 Results

Table 5.2 shows the accuracy and the AUC of all four experiments. As a benchmark, always predicting the car will result in an accuracy of 81% and an AUC of 0.50. The random forest model shows to have the highest performance, accurately predicting the mobility choice of 97% of the transactions in the last experiment. Generally, the accuracy and the AUC of all models in all experiments is relatively high, with minima of 88% and 0.91, respectively. The random forest outperforms the other models in most experiments, closely followed by the gradient boosted trees.

As expected, the performance of the models is high in Experiment 1 and decreases in Experiments 2 and 3. This can be explained as in the first experiment, unrealistic start and end locations and repeating choices help the models boost their performance. In Experiment 4, the performance improves, showing the relevance of the external data sources. Especially the random forest and the gradient boosted trees benefit. Surprisingly, the performance of the neural network decreases in Experiment 4 and achieves lower performance than the logistic regression. This might be attributed to overfitting and can possibly be prevented by a more advanced parameter selection procedure.

Figure 5.7 shows the feature importance for the random forest model of Experiment 4. On the left (a), the mean absolute importance is shown. Interestingly, all features related to the reachability of the neighborhood are important. Hereafter, the indication for commute has a large impact. The specific travel times (time PT, time car) show to be relatively unimportant in comparison. Congestion also seems to have relatively little effect, as well as features corresponding to weather conditions.

TABLE 5.2: Experimental results: accuracy and (*AUC*).

Model	Exp. 1	Exp. 2	Exp. 3	Exp. 4
Binary Logit	98% (*0.99*)	91% (*0.91*)	88% (*0.92*)	89% (*0.93*)
Logistic Regression	98% (*0.99*)	91% (*0.91*)	91% (*0.93*)	91% (*0.94*)
Neural Network	98% (*0.99*)	95% (*0.97*)	93% (*0.96*)	88% (*0.92*)
Gradient Boosted Trees	98% (*0.99*)	94% (*0.95*)	94% (*0.91*)	97% (*0.99*)
Random Forest	99% (*0.99*)	95% (*0.97*)	92% (*0.95*)	97% (*0.99*)

On the right (b), the impact of all feature values on the model outcome is shown. A negative SHAP-value (negative impact on the model outcome) implies that the prediction tends to go towards public transport, a positive value towards the car. The more extreme the SHAP-value is, the higher the impact of the feature on the model output. The results confirm our intuition, a low ratio (car over public transport) of reachability results in a large negative impact. A low reachability by public transport or a low travel time by car result in a large positive impact. Interestingly, personal trips are preferred by car and commute trips by public transport. Also, a low commuting distance implies a preference for public transport. Lastly, congestion seems to have a positive model impact, implying a slight preference for cars when roads are congested.

5.6 Discussion and Conclusion

Our results show that the travel behavior in our dataset is highly predictable, as we can predict the individual transactions with an accuracy of 97%. Compared to the benchmark of 81%, this is a significant increase. Additionally, by quantifying the importance of all features, we show insights into why travel behavior is predictable.

The main features contributing to the models are our proposed reachability features. This conforms to our intuition; however, surprisingly, their importance is much higher than the specific travel times of the specific transactions. The general reachability of an area is more important for modal choice than the specific reachability. General reachability refers to the reachability of the neighborhoods, specific reachability to that of the specific trip the person is planning. People might not take the effort to check the travel times for the alternative and decide based on their general knowledge of the destination (neighborhood) reachability. This insight can be seen as an opportunity to inform individuals to stimulate behavioral change proactively.

Besides the reachability, the travel type has a considerable influence on travel mode choice. Commute trips are favored by public transport but private trips by car. This might be influenced by the travel policy of the company or the ability to work during public transport. The CO2 emissions of public transport also have relatively high importance. However, this might be interpreted by the model as an indication of whether the transaction contains bus trips. In the Netherlands, the train, metro, and trams are relatively low on CO2 emissions. The emissions are high only when transactions would involve the bus.

The features concerning congestion and weather are of little influence. The congestion might be explained as roads are typically congested at the start and end of a

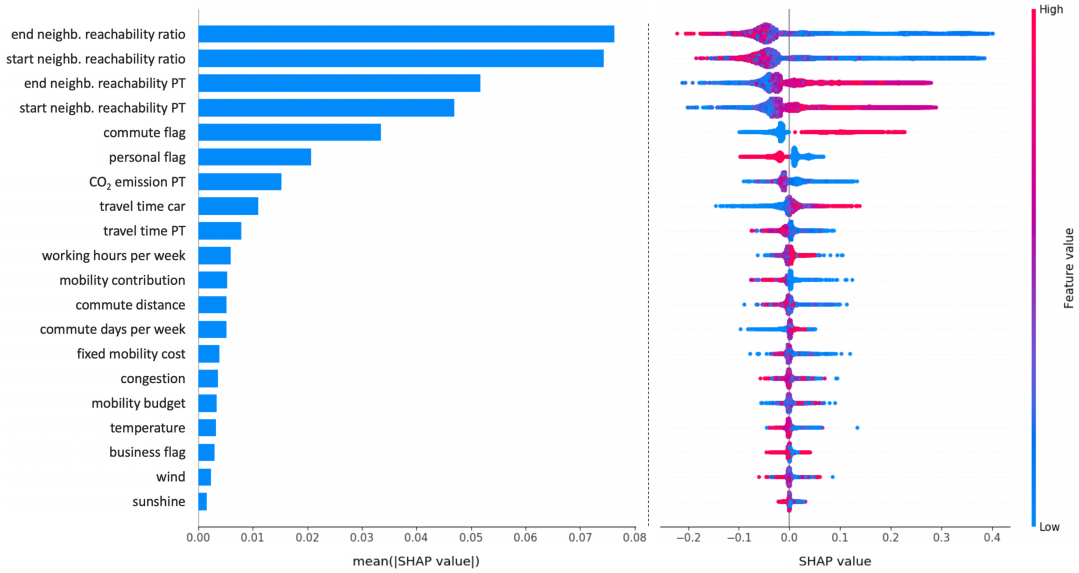


FIGURE 5.7: Feature importance: (a) mean absolute importance; (b) all feature values.

working day, and all persons in our dataset are employed. The SHAP-values even indicate that high congestion corresponds to a positive impact on model output, meaning a higher probability of taking the car.

Our experimental setup shows that data processing is critical for the evaluation of the models. If we would simply only execute the first experiment, we could present models with even higher accuracy. However, they would explain modal choices in a limited fashion. For example, public transport transactions can be predicted easily as their start and end location are at stations and travel times between stations are relatively fast by public transport. Additionally, the experimental setup highlights differences between the models. In the first experiment, all models perform similarly, however, in the final experiment differences are clearly visible.

5.7 Research Opportunities

The models developed in this research show promising results and give insights into the mobility behavior of the individuals. Still, they can be improved and used for further purposes.

Firstly, the models can be used to predict the impact of changes in infrastructure. The mobility behavior of the individuals is incorporated into the models. When the infrastructure changes, the features in the data change, and the mobility choices of the individuals might as well. Our model can be used to quantify to what extent investments in infrastructure lead to different mobility behavior.

Next, we can introduce more specialized models to gain more accurate predictions. Specifically, the availability of alternatives and repeating choices can be incorporated explicitly in a model. The advantage is that we can use more data to fit the model, as currently, we filter the data on these conditions.

Additionally, we can introduce trip chaining. This incorporates the fact that trips of individuals are linked throughout time and possibly influence each other. For

example, if an individual first needs to take their children to school and after that directly go to work, the modal choice for the trip to work is influenced by the trip to the school. In the data presented in this research, we can follow the choices of an individual throughout the day, hereby combining the trips that need to be executed throughout the day. For example, if one of the destinations throughout this day has historically only been executed by car, we need to take this preference into account for the other trips during that day.

Furthermore, the estimation of start and end locations can be further improved. At the moment, we take a constant travel time of ten minutes by bike and consider the buildings in the surrounding area weighted by population density. However, the willingness to bike might vary per station and region. For example, the density of stations in cities varies, which possibly has an impact on the willingness to bike. Also, it might be more relevant for business trips to weight the buildings by the number of employees.

Also, this research can be extended towards influencing the travel mode choice of individuals. We can use the predictions of the model to compare individuals amongst each other, and amongst the expected behavior. If the choices of an individual differ from a cluster or the expectation of the model, this might be an indication that the behavior can be changed. This potential change can be communicated easily on an individual level through a mobile application.

Finally, we can investigate whether we can influence the travel time of individuals. We know the expected choices of individuals and we know the expected choices of our whole population. We can combine these to spread the traffic flows on multiple travel modes. This in order to minimize congestion or the stress on a system. Especially during crisis situations, such as the Covid-19 virus, such an extension could be relevant.

6 Accessibility Analysis for Private Car and Public Transport: Comparable Measures for Data-Driven Policymaking

6.1 Summary

The disparity between the accessibility of areas through different travel modes is essential for the choice of the mode of transport. Calculation of the travel times by different travel modes is, therefore, very important. Many urban design decisions on infrastructure depend on these calculations. Developments in open data policies among urban data producers make this analysis more tractable. In this chapter, we apply a data-driven approach to travel time estimation based on realized past travel times. We compare commuters who drive in a car with those who use the train in the same geographic region of the Netherlands. First, we propose a method to quantify the accessibility of areas for these different modalities. Second, we show how these metrics can be used to determine optimal locations based on the willingness to travel. The results can be integrated into planning software to making data-driving decisions for policymaking.

6.2 Introduction

Many advanced traveler information and transportation management systems depend on an effective prediction of the accessibility of geographical areas. The analysis of accessibility is essential to study the interaction between transportation and land use [12, 117]. Traditionally, accessibility has been calculated using the privately-owned car as the subject. However, recent concerns on the environmental and social sustainability of land use warrant the need to incorporate different modes of transport in the accessibility analyses.

A significant disparity in accessibility over different modes of transportation can have a major impact on equality in society. Several studies point out that many urban regions in the US and Europe provide better levels of access when using the private car instead of public transport [57, 72, 73, 80, 116, 117]. Therefore, people who are not driving financial, physical, or lifestyle-related reasons may face difficulties accessing services and opportunities [91].

Travel time estimation can be done in several ways. For transportation by car, many navigation systems use GIS software where road segment lengths are divided by their corresponding speed limits providing estimates of the free-flow drive time. The accessibility of areas is then reduced to the shortest path problem between the

origin and destination locations. This calculation disregards potential congestion. This factor may significantly alter the travel time in an urban setting [27, 90, 154]. Studies have been published in which the travel time calculation is adjusted based on congestion (see, e.g., [57, 85]). The manner in which this is done is not reported in detail.

Travel time estimation for public transport comes with different challenges. Public transport is restricted to predefined routes and schedules that are time-dependent. Assumptions relating to travel speeds along the route and the transfer times between different lines may impact the correctness of the calculations [79]. Some studies take all stages of the journey into account from the origin to destination location (e.g., walking to the public transport station, waiting times for arrivals, transfer times, and walking times from the destination station to the final destination) [11, 79, 82]. Nowadays, this information is available through electronic journey-planning systems through APIs to provide such data for planning purposes.

This chapter applies a data-driven approach to travel time estimation based on realized past travel times. These travel times are, therefore, time-dependent. We compare commuters who drive in a car with those who use the train in the same geographic region of the Netherlands. First, we propose a method to quantify the accessibility of areas for these different modalities. We define an area, compute travel times, and finally propose a method to combine these into a comparable metric. Second, we show how these metrics can be used to determine optimal locations based on the willingness to travel. The results can be integrated into planning software to making data-driving decisions for policymaking.

6.3 Methodology

In this section, we highlight the methodology for tackling our two research questions. First, we quantify the accessibility of areas for different modalities. Second, we highlight the placement of physical locations to maximize the extracted potential from a network.

6.3.1 Accessibility of Areas

In this subsection, we propose a method to quantify the accessibility of areas for different modalities. This is challenging as it heavily depends on the infrastructure, the direction, and the demand for travel. These factors vary for different areas. To tackle this, we first define an area, then compute travel times, and finally propose a method to combine these into a comparable metric.

Defining areas appropriately is critical, as this heavily impacts the computational effort of our method. Besides, it allows us to incorporate the demand for travel. Computing travel statistics between coordinates on a granularity level of centimeters would require too much effort. Therefore, we aggregate our scope to so-called areas. We do so by taking the definition of neighborhoods of the Statistics Bureau of the Netherlands (CBS) [22]. These neighborhoods are similar to a postal code classification. They are based on the population and the economic density. Hence, they also give an indication of travel demand. The neighborhoods are maintained and updated every year.

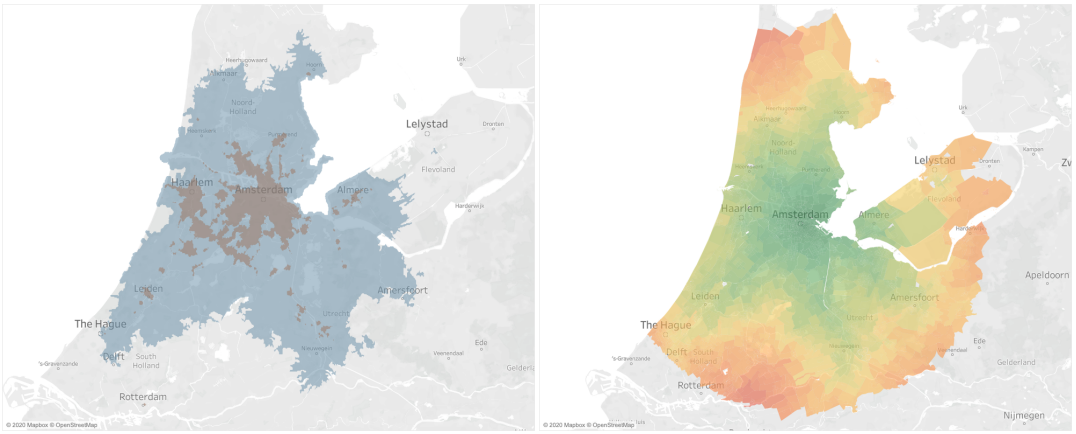


FIGURE 6.1: Accessibility from Amsterdam: (a) shapes accessible within 50 minutes by car (blue) and 60 minutes by public transport (grey); (b) travel time to 4,000 closest (straight line distance) areas by car.

After defining an area, we need to compute travel times between them. The challenge of using an area (shape) instead of a coordinate is that, in the end, we still need a pair of coordinates to compute travel statistics. We tackle this challenge by defining a coordinate to represent each area. We define this coordinate as the building located closest to the center of the area. We cannot simply take the center of the area, as this might be in the middle of a park, body of water, or in a forest. The dataset containing the location of all buildings in the Netherlands is available upon request by the government through [70].

Computing travel statistics between coordinates can be done using various services through their APIs. Depending on the rate limits of the chosen service, this can be done for free. We used [143] to compute travel times and distances between most areas. Figure 6.1 visualizes an example of travel times starting from the Olympic Stadium in Amsterdam. On the left (a), the shape which is accessible by car (blue) and public transport (grey) is shown. The travel time used for car is 50 minutes, and for public transport 60 minutes. The difference is to take into account the time required for parking. On the right (b), the figure visualizes the travel time to 4,000 closest areas (straight line distance) by car. Green represents a short travel time, and red indicates a long travel time.

Having the definition of an area and structure for computing travel times in place, we can quantify the accessibility. We do so by computing the travel time to all neighboring areas within a certain range, calculating the velocity to each neighboring area, and finally averaging all velocities. The range can be varied depending on the use case (short versus long-range accessibility). We see the calculation of the velocity as a different step, as we take into account the distance as the crow flies. This allows for a better comparison amongst modalities, as the required travel distance depends on the infrastructure of the corresponding modality. By taking the average over all velocities, we automatically emphasize velocities to high-demand areas, as these areas are smaller compared to low-demand areas. The resulting number represents velocity (km/h), which can be compared amongst different areas and modalities.

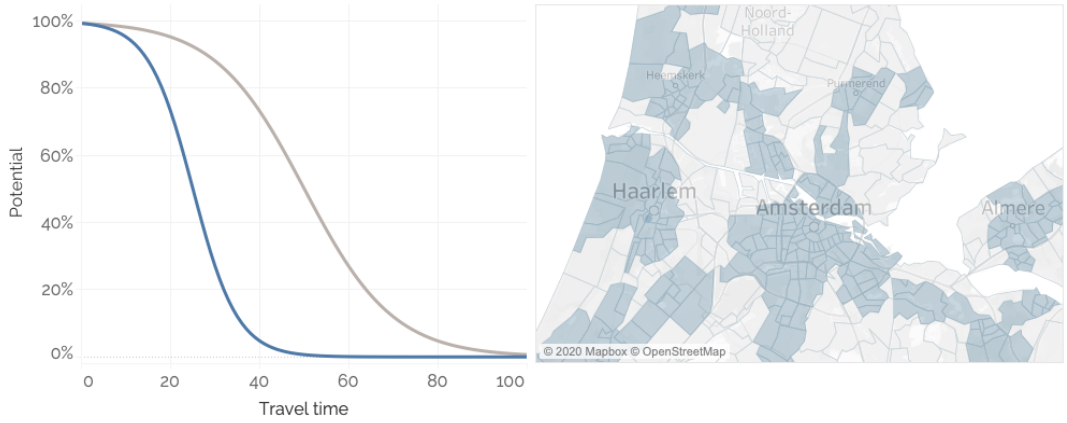


FIGURE 6.2: Defining willingness to travel: (a) travel time (minutes) versus the extracted potential for densely (blue) and sparsely (grey) populated areas; (b) a map of densely (blue) and sparsely (grey) populated postal codes in the Netherlands.

6.3.2 Placement of Locations

The second problem we are considering is the optimal placement of physical locations: maximizing our achieved network potential with a fixed number of locations. We do so while considering cannibalism and regional differences of various sorts. The locations can be of various types, such as stores, offices, stations, or dealerships. The potential can be broadly defined but generally depends on statistics like the population density, the current sales distribution, or the purchasing power. We consider the placement in aggregated areas, defined similarly to postal codes. However, the method proposed is independent of this choice. In this subsection, we explain how we compute the potential of a network, and we define a greedy algorithm for finding a lower bound on an optimal placement.

Before we illustrate our approach, we must define the willingness to travel. We do so by defining a curve that specifies the relation between the travel time to a location and the percentage of achieved potential. An example of this curve, a sigmoid curve, is shown in Figure 6.2(a). Generally, a location will achieve all of its potentials if the travel time is short, and little if the potential if the travel time is long. The specific relation depends on the purpose of the analysis. It can be derived by using an analysis of current customers or by using surveys. Additionally, the curve depends on the location itself. For example, people living in sparsely populated areas might be willing to travel longer compared to people living in densely populated areas. In this research, we consider applying different curves depending on the population density of an area. Figure 6.2 (b) visualizes different classifications of population density on a map of the Netherlands.

Our approach starts by defining three input tables. These tables contain basic statistics that are required for any approach to finding an optimal placement. A sample of each table is shown below. Table 6.1 contains the travel times between postal codes and a curve identifier. These travel times are computed as in Section 6.3.1. The curve identifier specifies which curve is applicable for the corresponding postal code (*to_pc*). Table 6.2 specifies the relationship between the travel time and the maximum potential (percentage) that can be achieved. Multiple curves for different

TABLE 6.1: Travel time.

pc	*pc_to*	*travel_time*	*curve_id*
1000	1001	10	1
1000	1002	12	1
1001	1200	40	2

TABLE 6.2: Travel time curve.

travel_time	*curve_id*	*potential*
0	1	100%
15	1	50%
15	2	30%

TABLE 6.3: Potential.

pc	*potential*
1000	20
1001	25
1002	18

TABLE 6.4: Sample of the resulting table. Postal code is abbreviated with 'pc' and potential with 'po'.

pc	*pc_to*	*travel_time*	*curve_id*	*po_raw*	*po_sum*	*po_max*	*po_correction*	*po_achieved*	*po_available*	*po*
1000	1001	10	1	80%	160%	80%	0.50	40.0%	25	10
1000	1002	12	1	80%	170%	80%	0.47	37.6%	18	6.8
1000	1400	40	2	10%	80%	50%	0.63	6.25%	10	6.3
1200	1001	28	1	50%	160%	80%	0.50	25.0%	25	6.3
1200	1002	20	1	55%	170%	80%	0.47	25.9%	18	4.7

postal codes can be considered. However, each postal code can only have a single curve. Table 6.3 contains the absolute potential per postal code. This potential can represent expected sales and can, for example, be computed by an analysis of the current sales in combination with demographics such as population density, income, age, family composition, or retail activity.

Using these tables as input, we compute the total sales potential and network coverage in the following way. As a basis, we filter Table 6.1 on *pc* to only containing postal codes of locations in our considered network. Next, we left join Table 6.2 on *travel_time* and *curve_id* to convert travel time to potential (relative). If looking at a single location, this potential would represent the achieved potential. However, when considering multiple locations, we need to correct for cannibalism. If we would not do so, we could place n locations in a single postal code and achieve n-fold the potential of this postal code.

Correcting the potential for cannibalism can be done by computing two statistics from our intermediate table: the summed potential (*potential_sum*) and the maximum potential (*potential_max*) per postal code. We do so by grouping the table created in the previous paragraph by postal code (*pc_to*). It is clear that the summed potential of a postal code cannot exceed 100%. The maximum potential, however, is less intuitive to take into account. It describes the potential that the most nearby location tries to achieve. We argue that the summed potential cannot exceed the maximum potential. An example might best illustrate why.

Suppose we choose one location, location 1, in postal code a. We only consider one different postal code from which we can get potential, postal code b. Suppose the travel time between them is 20 minutes, resulting in an achieved potential of 50%. In this scenario, location 1 would get 50% of the potential from postal code b. Now consider a second scenario by placing two additional locations (locations 2 and 3) in postal code a. Locations 1, 2, and 3 all want to extract 50% potential from postal code b. The summed potential would be 150%. We could correct this solely on the summed potential by capping it to 100%. Locations 1, 2, and 3 will each get 33% potential by doing so. However, we think this overestimates the achieved potential. From the perspective of postal code b nothing has changed. The travel time to the closest location is in both scenarios is 20 minutes. However, the latter would have twice the achieved potential. Thus, we propose that the summed potential cannot

exceed the maximum potential. In scenario 2, locations 1, 2, and 3 will each achieve 16.7% potential.

After correcting for cannibalism, we can finalize the computation. We left join Table 6.3 on postal code (pc), multiply the achieved potential with the absolute potential, and sum the resulting potential column. An example of the final table is shown in Table 6.4.

Algorithm 1 Locations Selection

Input: L
Output: L^*
$L_0^* = \emptyset$
for $i \in \{1, \ldots, N\}$ **do**
  $L_i^* = \text{find\_next\_best}(L, L_{i-1}^*)$
  for $l \in L_i^*$ **do**
    $\bar{L} = L_i^* \setminus \{l\}$
    $L_i^* = \text{find\_next\_best}(L, \bar{L})$
  end for
end for

Algorithm 2 Find Next Best

Input: L and L^*
Output: L_{new}^*
$p^* = 0$
for $l \in L \setminus L^*$ **do**
  $L_{new} = L^* \cup \{l\}$
  $p_{new} = \text{calculate\_potential}(L_{new})$
  if $p_{new} > p^*$ **then**
    $p^* = p_{new}$
    $L_{new}^* = L_{new}$
  end if
end for

Now that we can compute the potential of a given network, we can find an optimal network. Given a fixed number of locations, we want to maximize the potential. We define a greedy algorithm, which finds a lower bound on an optimal placement. The algorithm is a constructive algorithm that chooses one location in each iteration until the number of required locations is reached. Algorithm 1 describes this procedure. We are considering M possible locations, from which we need to pick $N \leq M$. The input is the set of locations $l_1, \ldots, l_M \in L$, and the output is the set of chosen locations L^*. The set L_i^* is defined as the locations chosen in iteration i, $i \in 1, \ldots, N$.

6.4 Results

In this section, we highlight our most important findings from analyzing the accessibility of areas and the placement of locations.

The accessibility of neighborhoods in the Netherlands is visualized in Figure 6.3. On the left (a), it shows the accessibility by public transport; on the right (b), it shows the accessibility by car. Both graphs clearly indicate the location of the existing infrastructure. Neighborhoods containing major train stops or highways are indicated green. On the other hand, neighborhoods that contain hardly any infrastructure are indicated red. More notable are the differences between the graphs. Public transport has a relatively large number of areas with low accessibility.

Computing the accessibility on a neighborhood level allows us to zoom in on specific areas and compare modalities. Figure 6.4 does exactly that for Amsterdam. We observe large differences between public transport and car. Public transport shows relatively high accessibility in the city center and relatively low accessibility outside the city center. Regarding the car, it intuitively is the other way around: high accessibility surrounding the city center, but not inside.

Figure 6.5 visualizes the output of the location placement procedure in two ways. On the left (a), by plotting an instance of the achieved potential per postal code when

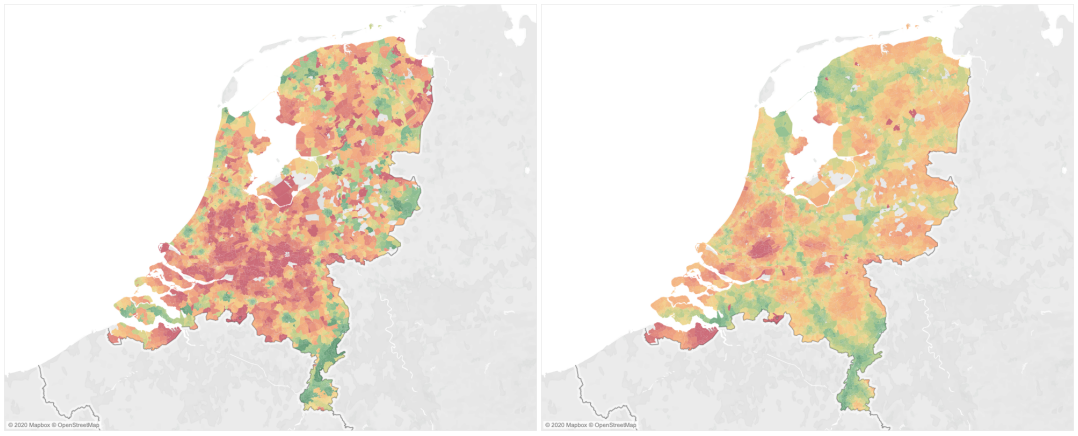


FIGURE 6.3: Quantifying accessibility of the Netherlands by: (a) public transport; (b) car. Relatively low accessibility is colored in red, high in green.

choosing five locations in the Netherlands based upon population density. Postal codes from which we extract a high potential are colored green, postal codes from which we extract a low potential are colored red. On the right (b), by highlighting the relationship between the number of locations placed and the total achieved potential. We observe that increasing the number of locations also increases the achieved potential. However, there are diminishing returns. In our instance, we achieve 80% using just ten locations. Yet, we gain only 12% more potential by placing an additional fifteen locations. Extracting 100% of the network potential will require many more locations.

6.5 Use Cases

The methodology developed in this research can be used for various use cases. In this section, we highlight two: (1) placing mobility hubs and (2) placing retail buildings.

First, the accessibility of areas is especially useful when deciding on where to improve existing infrastructure. This can, for instance, be improved by placing mobility hubs. In these hubs, travelers can switch between modalities or share mobility services. These hubs need to be placed in locations relevant to potential customers, i.e., where demand is high, but supply is low. We can quantify demand in terms of population density and supply in terms of our computed accessibility. The relation between different modalities is also interesting; for instance, we can find areas with high car accessibility but low public transport accessibility. By comparing these figures, we can decide on where to place hubs quantitatively.

The methodology related to the placement of locations is especially powerful as it can be used in various use cases. All use cases must have the characteristic that travel time is a factor that influences the demand for a location. We can modify our input tables to suit the specific needs, for example, when placing retail stores. First, we can estimate the potential per postal code by analyzing the current customer base, for instance, by looking at a certain age or income range. Next, we estimate the willingness to drive. Depending on the store type, people might consider a certain travel time or transport type to visit the store. We can fit a sigmoid curve on the

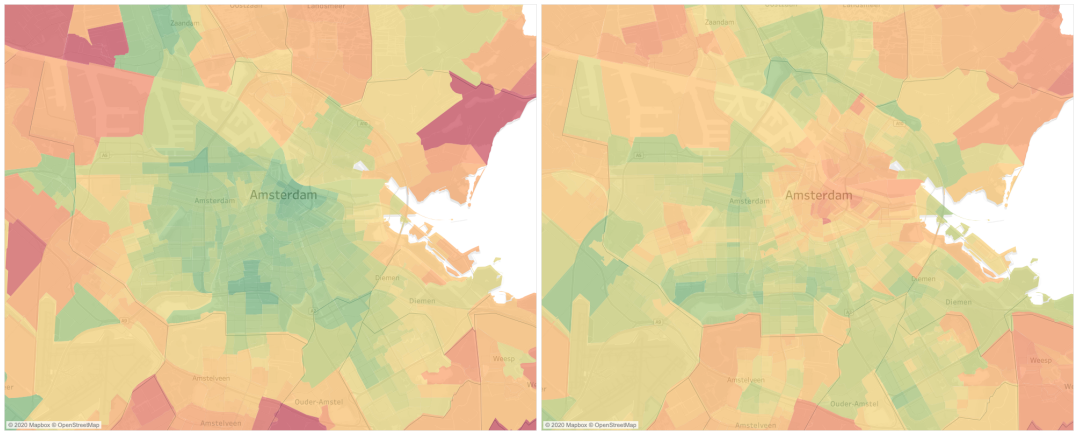


FIGURE 6.4: Quantifying accessibility of Amsterdam by: (a) public transport; (b) car. Relatively low accessibility is colored in red, high in green.

travel time of their preferred transport type. The locations considered might vary on the current network, competitor networks, and a rough initial selection of available locations. The travel time matrix remains intact regardless of the use case. Next, we follow the methodology as described in Section 6.3.2 to get advice on the placement of the stores.

6.6 Discussion

In this chapter, we have shown methodology for quantifying the accessibility of areas and placing physical locations to extract a maximal potential from a network. In this section, we discuss our findings and highlight potential improvements to our methodology.

Regarding the computation of the accessibility, we observe that postal codes around the border of the Netherlands show remarkable values. Typically, we see the accessibility lies in the extremes, it is either high or low. This can be explained by the fact that we only take into account neighborhoods within the same country. In Limburg (the most southern province of the Netherlands), for example, we compute the accessibility by taking into account mainly other neighborhoods in Limburg. Combining this with the fact that it is narrowly shaped and there are both a highway and a railroad running through it, the resulting accessibility will be relatively high. If there had not been a highway, the accessibility would have been low. We could improve our computations by also taking into account neighborhoods of neighboring countries.

Besides, the accessibility is currently computed only for postal codes containing buildings. We computed travel times between postal codes and defined the center as the building located closest to the geographical center of the area. However, if there are no buildings in a postal code, we do not define the center of the postal code and do not compute the accessibility. This results in a few white spots in our analysis. We could impute the values for these postal codes by taking the average accessibility of the surrounding postal codes, or redefining the center of them.

The methodology regarding the location placement is especially powerful, as the achieved potential of one network configuration can be computed using a single

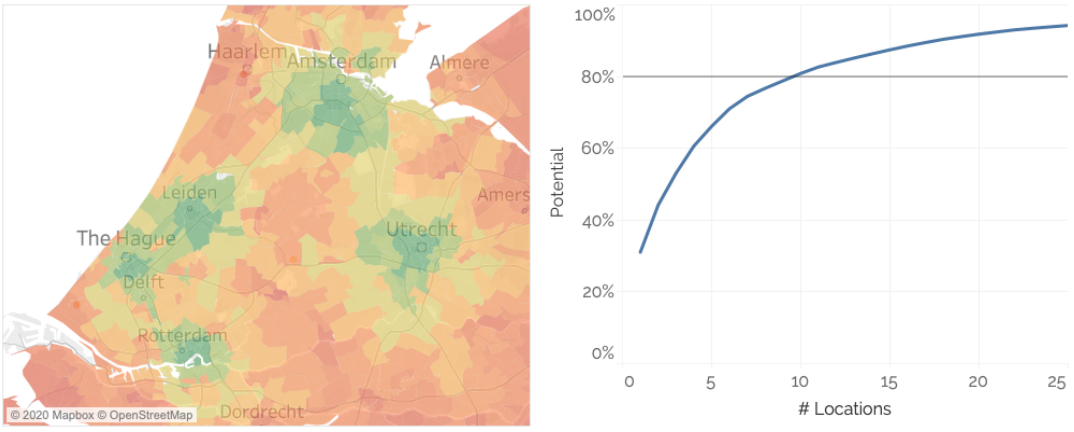


FIGURE 6.5: Location placement based upon population density: (a) extracted potential per postal code from when choosing five locations: high (green) versus low (red); (b) relation between the number of locations placed and the extracted potential.

query. This query can be executed using a commonly used language, like MySQL. This creates the opportunity to integrate the methodology into standardized tools. By doing so, users can interact with the methodology. However, as with our other methodology, it is not perfect. We see the main extension to include different forms of transportation. The willingness to travel might vary per modality, and potential customers might not have access to all modalities. Besides, we could attempt to include congestion in the travel time. The travel time we are using is currently computed by an API that does not take this into account. Also, we could refine the location placement by including various statistics regarding the cost of the location. This allows us to balance the trade-off between achieved potential and expenses.

Acknowledgments

The authors would like to thank Wilte Falkena for his help in creating part of the figures.

7 Overcoming the Self-Fulfilling Prophecy in Time Series Forecasting

7.1 Summary

Two current challenges in time series forecasting are the *self-fulfilling prophecy* and finding *robust seasonal patterns*. We argue that both can be overcome through combining similar time series. We propose methodology to extract robust seasonal patterns from low-level sales data through applying hierarchical clustering. We validate our approach using a simulation experiment and a real-life dataset containing over €2B of bicycle sales. Our simulation results show a 45% decrease in forecasting error and they quantify the effects of the self-fulfilling prophecy on forecasting error. Our results on real-life data show a 15% performance gain on the benchmark when applying clustering. Additionally, we show insights in the effects of applying smoothing and forecasting sell-in vs sell-out data.

7.2 Introduction

In this chapter, we focus on two challenges within time series forecasting: fitting robust seasonal patterns on volatile data and the self-fulfilling prophecy. We aim to overcome both challenges through applying hierarchical clustering based on the similarity of seasonal patterns. We have access to a unique, confidential dataset consisting of over €2 billion of bicycle sales in western Europe to validate our approach.

To illustrate the first challenge, we take the bicycle industry as an example. Their sales data suffers from large process influences. The model lineup, production planning, and component availability each introduce volatility in sales data. The model lineup causes a high turnaround in products; over 50% of 2020 sales is e-bikes, these did not exist a few years ago. Production causes sales peaks after production batches, as current demand exceeds supply. Component availability causes the production of some products to be temporarily impossible, as suppliers are facing downtime due to Covid-19 lockdowns. Combined with a yearly seasonal period this makes it difficult to estimate accurate seasonal patterns based on sales data. Forecasting algorithms need multiple periods to estimate the seasonal component accurately, especially if the data is volatile. Thus, algorithms will only be able to forecast demand with reasonable accuracy after two to five years. For decision makers, this is too late.

The second challenge is the self-fulfilling prophecy. Simply put, it means that the companies will work hard to make their own predictions come true. Through demand planning, a company decides on how much product it expects to sell. Next, production ensures this amount of products will be produced, and sales ensures this amount will be sold. At the end of the year, all their decisions will show up in the sales data. This sales data is once again used for estimating demand for the next period. And so the cycle continues. Yet, there is a difference between sales and demand. What if your initial forecast was off, and you are creating your own truth?

An attempt to tackle both problems is to combine data from related time series. Possibly, one can fit seasonal patterns more rapidly and find the underlying demand more accurately. When limited data is available across across multiple periods, the potential of combining similar series within the same period might lead to more data. Little data in series, more data in parallel. Various approaches have been proposed to combine data when predicting a single series, mainly within the fields of hierarchical clustering and cross-learning.

7.2.1 Hierarchical Forecasting

Time series can often be represented in a grouped or hierarchical structure. This structure might be geographical, for example, sales in a city can be aggregated to region, country, continent, and the world. The structure can also be a product dimension. Hierarchical forecasting exploits this structure to generate better forecasts. A main challenge is to compute accurate forecasts which are coherent across the aggregation structure. That is, the sum of low-level forecasts (e.g., countries) must add up to higher-level forecasts (e.g., continent). In addition to the ability to provide coherent forecasts, hierarchical forecasting has the potential to (1) improve forecast accuracy, and (2) reduce the magnitude of the forecasting problem [39].

Multiple approaches to hierarchical forecasting have been developed. Many of those are top-down or bottom-up approaches, or a combination [62]. The top-down approach forecasts on the highest aggregated level, and distributes to the lowest disaggregated level through applying historic proportions. The bottom-up approach forecasts on the lowest disaggregated level, and sums all the way to the highest aggregation level. The so-called middle-out approach does both: it choose a middle ground and disaggregates down through historic splits and aggregates through summing within the hierarchy. Typically, the lowest disaggregated level is volatile and hereby difficult to forecast, and the highest aggregated level smooth and easier to forecast. Both methods, however, show disadvantages. The bottom-up approach is not ideal as it might be error-prone [48, 38] and the top-down approach is not ideal because of information-loss [34, 59].

More recently, approaches have been proposed which reconcile multiple forecasts at different levels within the hierarchy. The goal of these is to produce better forecasts than either a top-down or bottom-up forecast. The optimal combination approach forecasts all series at all levels of the hierarchy and optimally combines those to a reconciled forecast [65]. This research is followed by the minimum trace (MinT) reconciliation approach [149]. A slightly different approach to hierarchical forecasting is proposed in [35] through introducing the game-theoretically optimal reconciliation (GTOP) method.

However, the hierarchical or grouped structures within hierarchical forecasts are typically made for political or business reasons, not for forecasting purposes. This might lead to imbalanced hierarchies (e.g., both the USA and Monaco are countries, however, the USA has over 8000x more inhabitants) and a large distance between potentially highly correlated product categories (e.g., popcorn and movie tickets). Therefore, we take the methods for hierarchical forecasting as an inspiration, however, we will propose a slightly different approach.

7.2.2 Cross-Learning

Cross-learning models are trained across an entire time series dataset in order to extract information from multiple series and accurately predict individual ones [113]. Especially when data is limited, sparse, or highly correlated, it can improve forecasting performance [87]. Various approaches have been proposed and tested on empirical data. The M competitions can be seen as a measure on how well forecasting methodology holds up against real-life data sets. These competitions have been designed with the goal to learn from empirical evidence on how to improve forecasting [89]. Cross-learning methods score high in this open-source and transparent competition. The top three performing methods of the M4 competition apply some form of cross-learning [88], and all top-performing methods of the M5 competition do so as well [87]. This highlights the potential and usefulness of cross-learning.

The methodology that we will propose in this research learns from multiple series in order to generate a forecast for a single series. Thus, it can be viewed as a cross-learning method.

7.2.3 Self-Fulfilling Prophecy

Sales are often confused for demand. However, there is a large difference. On the one hand, demand might be understated. For example, a bicycle manufacturer observes 0 car sales in their data. Yet, we know demand for cars does not equal 0. On the other hand, demand might be understated (constrained). For example, a manufacturer has a limited production capacity, thus, its sales cannot exceed this threshold. Process influences are a main driver for differences between demand and sales. This might have undesired effects in forecasting and might lead to a self-fulfilling prophecy.

This effect has been observed in different fields. In economic decision-making, evidence has been shown suggesting that speculative forecasts of economic change can impact individual's economic decision behaviour [98]. Within tourism cruise demand forecasting, research has confirmed the tendency of published forecasts on the market's development becoming self-fulfilling prophecies [75]. Additionally, it has been shown that an article published in The Economist containing a forecast on the Thai share price index seems a case of self-fulfilling prophecy, rather than one of good quality forecasting [58]. Even more severe, research has suggested that the use of forecasts to drive policy is potentially destabilizing [20]. Whereas much research has been devoted to historic sales on forecasts, little research has been devoted to the effect of forecasts on future sales.

7.2.4 Contribution and Outline

In this chapter, we propose methodology to extract robust seasonal patterns from low-level sales data through applying hierarchical clustering. We break the imposed hierarchy and create a new one based on similarity in the seasonal pattern of the decomposed time series. We validate our approach using industry data and a simulation experiment. Additionally, we provide insights on the effects of smoothing and sell-in vs sell-out sales data.

The outline of the chapter is as follows. First, we describe the real-life data that was made available under strict circumstances for this research. After, we describe the proposed methodology for generating forecasts. Next, we outline the experimental setup through which we validate our method. This section is followed by the results, and finally the chapter is wrapped up in the discussion.

7.3 Data

Under strict conditions, real-life data was made available for this research. This consists of bicycle sales data in western Europe spanning multiple companies. The time period ranges from 2015 until 2021, and the total revenue generated by these sales concerns over €2B. A common use case for this data is forecasting next year's sales on a model family level. A model family is a group of bicycles grouped by size, color, frame type, and component class. Our data consists of 84 model families.

Within this sales data, an important definition is the sales date. Despite recent developments in business to consumer (B2C) sales, bicycles are typically purchased at dealers in physical stores. Therefore, the products are sold twice: when the dealer makes a purchase from the brand, and when the customer makes a purchase from the dealer. These are called *sell-in* and *sell-out*, respectively. Both show different sales patterns and have their (dis)advantages. Our dataset contains both definitions, as we are interested in exploring their differences.

The raw data needs to be processed before we can use it for our analysis. First, we filter the data to contain a minimum sales volume of ten bikes per month. Second, we filter incomplete calendar years. For example, at the time of writing the year 2021 has not been fully accounted for, thus, we remove all 2021 data. After, we filter model families on having at least 24 consecutive months (two periods) of data available. Besides, as the focus of this chapter is on seasonal patterns only, we remove linear trends for each model family. We do so through fitting a linear model using the Python LinearRegressor class from scikit-learn [97] and subtracting the trend from the raw sales quantities.

Lastly, we want to investigate the usefulness of applying a smoothing technique to the sales data. As the sales quantities can be noisy at times, perhaps it might be useful to smooth them before fitting a forecasting model. To do so, we duplicate all our data and smooth all quantities on a model family level in the copied half. The approach used for smoothing is LOESS smoothing, implemented in Python through [19].

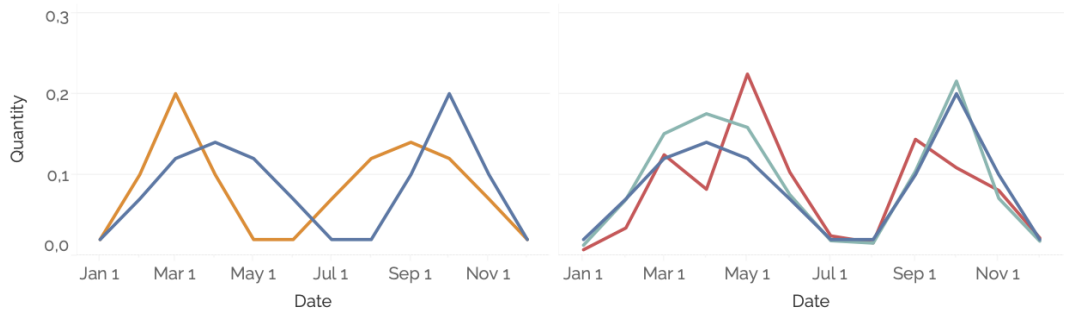


FIGURE 7.1: Seasonal patterns (left) and data types (right) considered for the simulation experiment. On the left, the default (blue) and reversed (orange) seasonal patter are displayed. On the right, demand (dark blue), sales (red) and forecast (light blue) are shown.

7.4 Methodology

We propose a relatively simple approach. We apply hierarchical clustering to all seasonal patterns extracted by any forecasting method on each series. The metric applied to determine the distance between two patterns is a the Weighted Average Percentage Error (WAPE). Comparable patterns will have a small distance and will be combined more quickly. After setting a distance threshold which determines the total number of clusters, we take a weighted average of all patterns in a cluster, weighted on sales volume. Next, we apply this extracted pattern on each series in the cluster.

Extracting the seasonal pattern from any forecast can be a challenge. Some methods explicitly model the seasonal component. In that case, we can directly extract it. In the other cases, one will have to de-trend the time series before applying the methodology. Typically, noise and the seasonal pattern will remain. This seasonal pattern can be extracted by, e.g., averaging the de-trended series within the period.

The clustering technique applied is agglomerative hierarchical clustering, implemented in Python using the *cluster.hierarchy* classes from [97]. As a linkage function, we use the complete (maximum) distances between all observations of two clusters.

The number of clusters is a parameter which needs to be set beforehand. In this research, we test five different parameter settings, on a logarithmic scale. These range from clustering all seasonal patterns to clustering no seasonal patterns. Cluster level 0 corresponds to merging everything; cluster level 4 to merging nothing. Levels 1, 2, and 3 lie on a logarithmic scale in between.

This approach shows parallels with cross-learning and hierarchical forecasting. The comparison with cross-learning is intuitive, as the clustered seasonal patterns contain information of various series combined. These pattern are hereafter used for the prediction of each of the single series. The approach also shows parallels with hierarchical forecasting. Essentially, we apply seasonal patterns in a top-down fashion on single series. However, we break the imposed series by the data and create a new hierarchy based on similarity of seasonal patterns.

We validate our approach using two experiments. The first concerns forecasting demand on the real-life data. The second concerns a simulation experiment.

Algorithm 3 Generating forecasts

```
result = ∅
data = read_data
data = process_data
for year do
  for data_type do
    for smoothing_method do
      for model_family do
        df = filter_data(year, data_type, smoothing_method, model_family)
        for forecast_method do
          method = forecast_method
          fcst = generate_forecast(df, method)
          result = append_forecast(result, fcst)
        end for
      end for
    end for
  end for
end for
result = cluster_forecasts(result)
```

7.4.1 Data Experiment

The experiment on the real-life data aims to investigate the effect of different dimensions on forecasting results. The most important dimension is the cluster size of the hierarchical clustering approach. Besides, we want to see if there is a difference between forecasting accuracy amongst different data types (sell-in vs sell-out), forecasting algorithms, and smoothing approaches. We do so through generating many different forecasts for all dimensions and evaluating the results.

The forecasting algorithms applied are ARIMA and Holt-Winters: these are well-known and widely-used algorithms. The parameters of both methods are tuned automatically to the data using their Python implementations *auto_arima* from [132] and *tsa.holtwinters.ExponentialSmoothing* from [112], respectively. Besides, we use last year's sales as a benchmark.

The experiment basically consists of generating one high-dimensional dataset by walking through multiple for-loops specifying the different dimensions. Each forecast should predict one period ahead, i.e., one full year. Creating a forecast requires the input of different dimensions: year, data type, smoothing method, and model family. We filter the data on all dimensions and generate a forecast using each forecasting method. This procedure is described in Algorithm 3. Once a forecast has been made, it is appended to a table specifying all dimensions and the forecasted quantity as columns.

The forecasts are evaluated by the WAPE, which is defined as:

$$\frac{\sum_{t=1}^{n} |A_t - F_t|}{\sum_{t=1}^{n} |A_t|},$$

with A_t being the sales at timestamp t, F_t the forecast at timestamp t, and n the length of data points to evaluate (typically only on the time dimension).

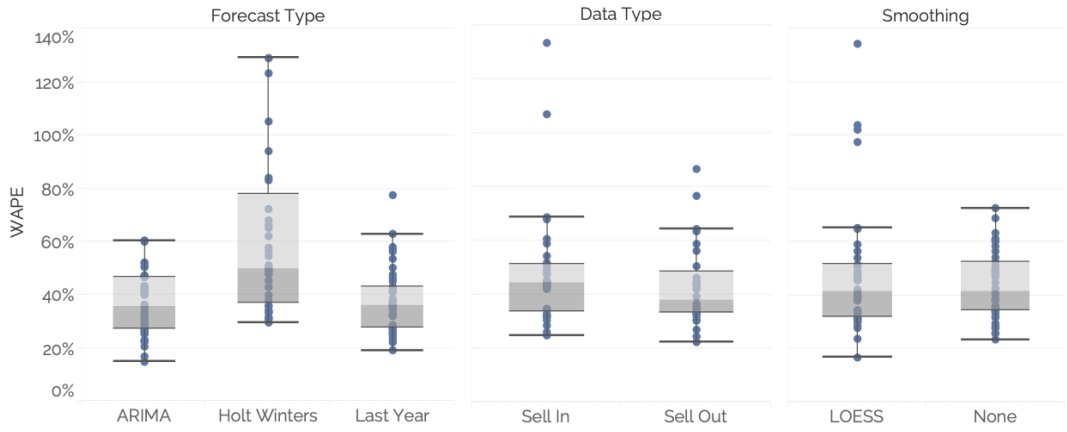


FIGURE 7.2: Results from forecasting the empirical sales dataset. Forecasting error (WAPE) on various bike models split by forecasting method (left), by data type (middle), and by smoothing method (right). Three data points are excluded from the Holt-Winters forecast type, its maximum WAPE corresponds to 310%.

To assess the impact of the different dimensions, we compute the WAPE per model family and the corresponding dimension. This allows us to compute a range per model family. Additionally, we compute the WAPE for each dimension value of the corresponding dimension. To derive these statistics from our resulting dataframe, we join the sales on year and month level, compute the absolute deviation, group the dataframe per model family and dimension, and compute the WAPE by dividing the sum of the absolute deviation by the sum of the weights (sales).

The clustering dimension is treated slightly differently; it is filtered out when evaluating the other dimensions. This because it might impose a bias towards clustering. We evaluate five clustering types, of which four cluster on different levels. If we would not leave out the clustering when evaluating the other dimensions, 80% of the evaluation would include some form of clustering.

7.4.2 Simulation Experiment

To validate the impact of the self-fulfilling prophecy on forecasting accuracy and to assess the usefulness of clustering in preventing it, we set up a simulation experiment. This is mainly motivated by the fact that using historic data we cannot quantify the consequences of having a different forecast. Besides, in real-life we never know the true demand underlying the sales. The experiment is simple in its basis. We generate N time series having no trend, a seasonal pattern of length 12 (monthly), and normally distributed noise with mean 0 and standard deviation of 0.5. The noise is applied in a multiplicative manner. For half of the series we apply the seasonal pattern in reverse. Figure 7.1 (left) visualizes the seasonal patterns.

We generate the time series in a year-by-year fashion. For each series, we generate three types of time series: (1) demand, (2) sales, and (3) forecast. The demand is equal to that of the seasonal pattern. The sales is a weighted average of the demand and forecast, controlled by a parameter $\rho \in \mathbb{R}$: $0 \leq \rho \leq 1$. The forecast is generated based on the sales. Actually, we generate two forecasts. The first forecast is simply the historic average of the corresponding series. The second forecast extracts all

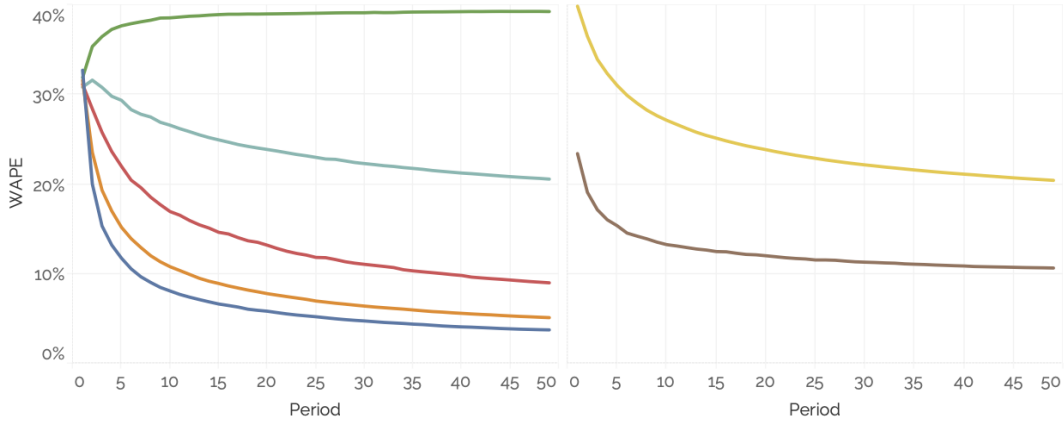


FIGURE 7.3: Simulation results: impact of process influences on forecasting error (left) and impact of clustering on forecasting error (right). On the left, different values for ρ are 100 (green), 75 (light blue), 50 (red), 25 (orange), and 0% (dark blue). On the right, the forecasting is done excluding (yellow) and including (brown) clustering.

seasonal periods by averaging, clusters them, and averages within the cluster. Figure 7.1 (right) visualizes a realisation of one period (year) of one series, consisting of the three different types.

We are interested in the impact of two main effects: the parameter ρ and the forecasting method. To do so, we measure the forecasting accuracy in terms of Weighted Average Percentage Error (WAPE) between the forecast and demand, weighted by demand volume. In total, we run ten simulations of 50 periods containing $N = 20$ series, with five values of ρ.

7.5 Results

7.5.1 Data Experiment

Figure 7.2 (left) visualizes the difference in forecasting error split by forecasting method. In order of increasing WAPE, these are ARIMA (35.8%), the last year benchmark (36.3%), and Holt-Winters (59.5%). Interestingly, the benchmark outperforms Holt-Winters and nearly the ARIMA model. The variance in error can possibly be explained by the data quality of the different model families over which the visualization was made.

Figure 7.2 (middle) highlights the impact of the data type (sell-in vs sell-out) on forecasting performance. The results show a large difference between the two, sell-in having a WAPE of 45.7% sell-out one of 38.6%. This is a relative difference of 18%. This seems intuitive, as the sell-out is less influenced by the processes created by the dealer and manufacturers.

Figure 7.2 (right) shows the impact of smoothing on forecasting performance. We observe little difference between applying and not applying LOESS smoothing. Applying smoothing lead to a WAPE of 44.4% and not applying smoothing to a WAPE of 43.1%. A few model families might benefit from smoothing, as the minimum WAPE for LOESS smoothing is lower than that of not applying smoothing. However, further investigation is necessary to draw conclusions.

Forecast method	Cluster level 0	Cluster level 1	Cluster level 2	Cluster level 3	Cluster level 4
ARIMA	35.7%	35.8%	35.7%	35.8%	35.8%
Holt-Winters	56.6%	57.7%	58.7%	58.7%	59.5%
Last year	31.0%	32.4%	34.0%	35.0%	36.3%

TABLE 7.1: Forecasting error (WAPE) split by cluster level and forecast method. A cluster level of 0 implies clustering the seasonal patterns of all model families, a cluster level of 4 implies no clustering. The levels in between scale down logarithmic.

Table 7.1 displays the results of applying hierarchical clustering on the seasonal patterns of the forecasts. The results differ per forecasting method. ARIMA shows hardly any difference, Holt-Winters shows a slight improvement, and the last year benchmark shows a large improvement when applying clustering. The benchmark even outperforms both the ARIMA and Holt-Winters model when applying any form of clustering. The performance boost in terms of WAPE can be up to 15% (relatively).

7.5.2 Simulation Experiment

Figure 7.3 (left) shows the impact of process influences on forecasting error. A high value for ρ implies a large process influence. We observe that the convergence towards a low forecasting error is much slower if the process influence is larger. For increasing values of ρ, the WAPE converges to approximately 4%, 5%, 9%, 21%, and 39% after 50 periods. Additionally, for the largest value of ρ, the forecast error converges towards a larger value than its initial value. It seems that the forecast and sales patterns settle on a different pattern that that of the demand.

Figure 7.3 (right) shows the impact of clustering on forecasting error. We observe a large difference between using and not using clustering. Firstly, the forecast using clustering converges much faster. After two periods with clustering a similar error is observed as without clustering after 50 periods. Secondly, it seems that the approach including clustering converges much faster. Even when accounting for the ten-fold increase in data available to fit seasonal pattern on. After five periods, the clustering approach as a WAPE of 15%, which is much lower than the 20% WAPE after 50 periods when not using clustering. After 50 periods of using the clustering approach, the WAPE settles at a value which is nearly twice as low (11% vs 20%).

Figure 7.4 displays two cluster dendograms within one simulation run at two points in time: after one period (left) and after 30 periods (right). The dendograms show large differences. After period 1, the distances between the extracted seasonal patterns of the products are much larger than those after 30 periods. Additionally, the distances within the clusters are much smaller after 30 years. However, the distance between the two clusters remains relatively large.

7.6 Discussion

Overall, we argue that this research shows promising results. On real data, our hierarchical clustering approach shows never to harm forecasting error. On the contrary, two out of three investigated forecasting approaches show a significant improvement. The results on the simulated data are even more striking, as forecasting

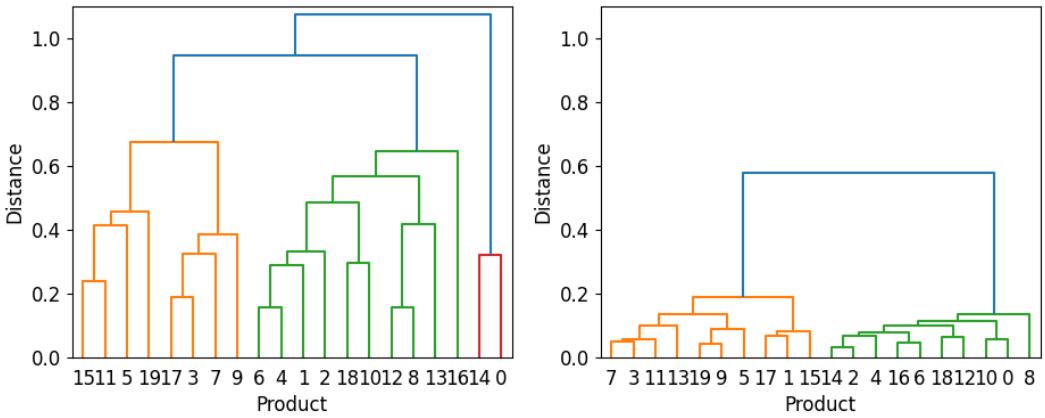


FIGURE 7.4: Cluster dendogram when linking seasonal patterns of all products in the simulation experiment: period 1 (left) vs period 30 (right).

error decreases by 45% when applying clustering. Additionally, the simulation experiment quantifies the large extent to which process influences impact forecasting fits.

The real data experiment shows one unexpected result, namely, that more clustering always improves performance. This is surprising, as we expected a trade-off amongst the cluster levels. More clustering should not necessarily lead to better seasonal patterns. Our results can potentially be explained by the noisy data in the bicycle industry. Besides, the ARIMA model does not seem to benefit from clustering. This can be explained by the challenges the model showed on fitting the data. The fitted ARIMA models show little variance in their predictions, which can explain why clustering does not change the results much.

The last year benchmark performs remarkably well in comparison with both statistical approaches. This might be explained by the self-fulfilling prophecy. The benchmark namely lies close to the approach the business uses to forecast their demand.

The results from smoothing the data seem to make no difference. Given that little data is available and the data is volatile, this is surprising. An explanation might be that smoothing smooths all peaks, including seasonal ones. Besides, different smoothing approaches with different parameters could have been chosen to further investigate.

An interesting finding from the simulation experiment is that there seems to be a self-reinforcing effect between forecast and sales. The speed-up in convergence exceeds the expected ten-fold increase, as data from ten similar products is available. This might be explained by the fact that forecasts close to the demand lead to sales data that is close to the demand. In the next year, this sales data is 'richer' compared to that of sales data generated partly by a poor forecast. A better forecast leads to better sales data, which in term leads to a better forecast, etc.

The simulation results also highlight the challenges faced in supply-driven markets. The Covid-19 pandemic has led to shortages in many supply chains globally, leading to markets changing from demand-driven to supply-driven. Pre-Covid, plenty of products and material were available, whereas post-Covid, sales are influenced by a

much greater extend on process influences (supply). In the bicycle industry, business experts estimate that this process influence can contribute to approximately 80% of sales.

Ideally, the scope of this chapter would have been on full forecasting predictions, not just on the seasonal pattern. However, we leave this for future research. For example, changes in trend make it challenging to compare and reconcile various products, leading to missed opportunities in clustering. Perhaps our methodology can be extended towards incorporating the full time series.

8 Approximate Dynamic Programming for Optimal Direct Marketing

8.1 Summary

Email marketing is a widely used business tool that is in danger of being overrun by unwanted commercial email. Therefore, direct marketing via email is usually seen as notoriously difficult. One needs to decide which email to send at what time to which customer in order to maximize the email interaction rate. Two main perspectives can be distinguished: scoring the relevancy of each email and sending the most relevant, or seeing the problem as a sequential decision problem and sending emails according to a multi-stage strategy. In this chapter, we adopt the second approach and model the problem as a Markov decision problem (MDP). The advantage of this approach is that it can balance short- and long-term rewards and allows for complex strategies. We illustrate how the problem can be modeled such that the MDP remains tractable for large datasets. Furthermore, we numerically demonstrate by using real data that the optimal strategy has a high interaction probability, which is much higher than a greedy strategy or a random strategy. Therefore, the model leads to better relevancy to the customer and thereby generates more revenue for the company.

8.2 Introduction

Customer communication is crucial to the long-term success of any business [126]. Research has shown communication effectiveness to be the single most powerful determinant of relationship commitment [115]. Companies can choose from multiple channels in reaching their customers. The recent rise of social media has expanded the possibilities immensely. Most research focuses on email communication, though, because it is relatively easy to collect data of every email sent and every interaction resulting from the email on a customer level. Therefore, a thorough analysis of email communication effectiveness is possible.

Currently, in most companies, domain experts determine the email strategy. Customers are selected for emails based on business rules. These rules can be deterministic, such as matching the email's language or gender with those of the customer. However, they can also be stochastic, such as matching the (browsing) activity categories of a customer to the email category. Measurements suggest that (1) a large fraction of the emails are unopened, (2) a larger portion of the emails do not even direct customers to the company's website, and (3) almost all emails are not related to direct sales. An increase in the interaction probability, therefore, directly leads to

additional revenue. This probability can be increased by a better recommendation process of deciding which email to send at what time to which customer.

The challenge faced in this research can be classified within the research field of recommender systems. A recommender system has the purpose to generate meaningful recommendations of items (articles, advertisements, books, etc.) to users. It does so based on the interests and needs of the users. Such systems solve the problem of information overload. Users might have access to millions of choices but are only interested in accessing a fraction of them. For example, Amazon, YouTube, Netflix, Tripadvisor, and IMDb use recommender systems to display content on their web pages [101]. Similarly, one can use recommender systems to recommend certain emails to users, thus, to determine when to send which email to which user.

Recommender systems have traditionally been classified into three categories: (1) content-based filtering, (2) collaborative filtering, and (3) hybrid approaches [2]. Content-based filtering is a recommendation system that learns from the attributes (or the so-called contents) of items for which the user has provided feedback [96]. By doing so, it can make a prediction on the relevancy of items for which the user has not provided feedback. Collaborative filtering looks beyond the activity of the user for which a recommendation needs to be made. It recommends an item based on the ratings of similar users [2]. Hybrid recommender systems make use of a combination of the above-mentioned techniques in order to generate recommendations.

Although recommender systems might seem a good way to address the direct marketing problem, they have some shortcomings. One of the major problems for recommender systems is the so-called *cold-start problem*. This concerns users or items which are new to the system; thus, little information is known about them. A second issue is that traditional recommender systems take into account a set of users and items and do not take into account contextual information. Contextual information might be crucial for the performance of a recommender system [3]. A third issue is an overspecialization: "When the system can only recommend items that score highly against a user's profile, the user is limited to being recommended items that are similar to those already rated" [2]. Lastly, recommender systems must scale to real data sets, possibly containing millions of items and users. As a consequence, algorithms often sacrifice accuracy for having a low response time [101]. When a data set increases in size, algorithms either slow down or require more computational resources.

The main contribution of this chapter is that we address the mentioned shortcomings of the traditional recommender systems by formulating the direct marketing problem as a Markov Decision Process (MDP). This framework deals with context and uncertainty in a natural manner. The context (such as previous email attempts) can be specified in the state space of the MDP. The uncertainty is addressed by the optimal policy as an exploration-exploitation trade-off. The scalability of the algorithm is addressed by limiting the history of the process to sufficient information such that the state space does not grow intractably large. Furthermore, we test our model with real data on a greedy and random policy as a benchmark. The results show that our optimal strategy has a significantly higher interaction probability than the benchmark.

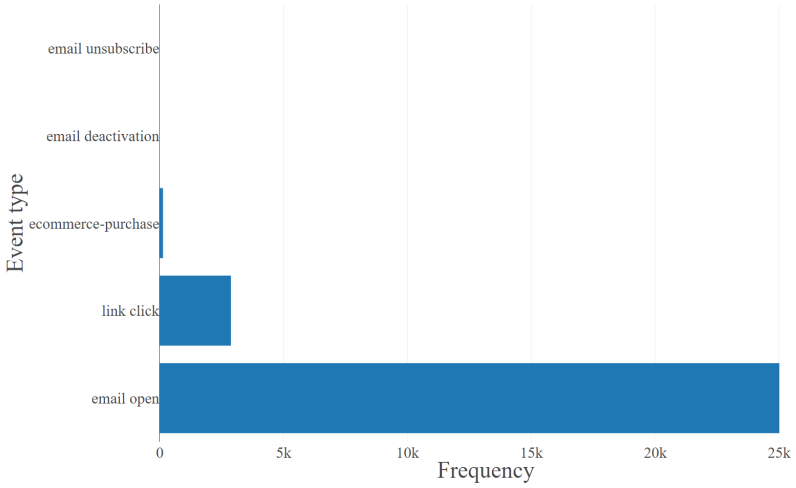


FIGURE 8.1: Frequency of event types.

In this chapter, we expand on our work in [126] by doing a more thorough data analysis, implementing an alternative method to solve the MDP, and by further elaborating on the discussion.

The organization of this chapter is as follows. In Section 8.3, we describe the data used for our data-driven marketing algorithm. Section 8.4 describes the model and introduces the relevant notation. In Section 8.5, we analyze the performance of the model and state the insights from the model. Finally, in Section 8.6, we conclude and address a number of topics for further research.

8.3 Data

In this section, we describe the data used for this research. We explain the data, comment on the data quality, filtering, and processing. Finally, we explore the data by showing relevant statistics and visualizations.

The data is gathered from five tables of an international retailer from one complete year and concerns: *sales* data, *email sent* data, *email interaction* data, *customer activity* data, and *customer* data.

The *sales* table contains all orders that have been placed by each customer. This includes information on the product, price, and date. The *email sent* table contains all emails sent to each customer. An email is characterized by attributes such as title, category, type, gender, and date. The *email interaction* table is structured similarly to the *email sent* table, however, it contains an interaction type. An interaction type can be email open, link click, online purchase, email unsubscribe, or email deactivation. The *customer activity* table contains for each customer its activity on the retailer's platform, such as browsing or clicking on the website. Finally, the *customer* table contains characteristics of a customer, such as date of birth, country, city, and gender.

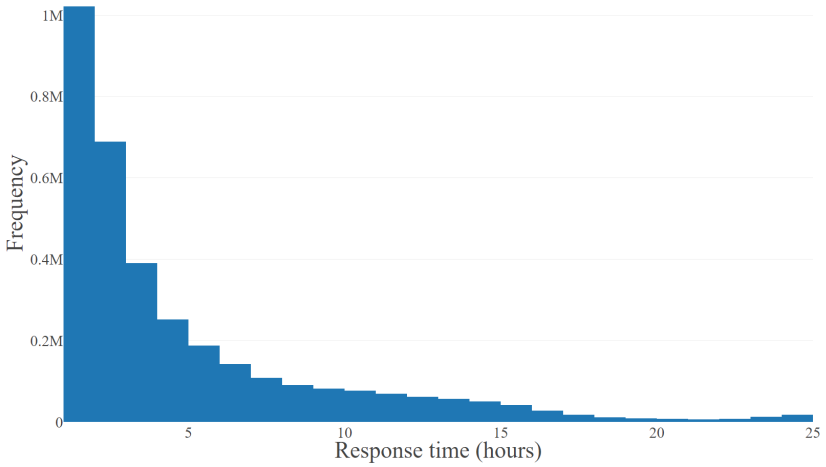


FIGURE 8.2: Distribution of time until the first interaction with an email.

Quality

The data used for this research is, for the large part, automatically generated. However, this does not guarantee its quality. Some issues appear when inspecting the data.

First, according to the data, 232 countries exist. Although there is discussion on the number of countries in the world, the United Nations (UN) recognizes a little under 200 countries. Business rules can explain the high number of countries in the database, such as classifying a part of the business (e.g., customer services) as a separate country. We tackle this issue by filtering on countries recognized by the UN.

Second, some physical stores are classified as individual customers. This results in these customers making hundreds of orders every year, creating much revenue. For these reasons, they can easily be identified.

Last, a large part of the customers does not place orders or show activity. This might be because one physical customer might have multiple accounts or devices through which interactions are made. Additionally, bots or spam accounts might be classified as customers. Business rules and logic is applied to identify and consolidate; however, this logic is not 100% accurate.

Processing

In this research, we analyze a vast amount of data. After filtering, we analyze approximately just over a million customers, but millions of emails, orders, and email interactions. Just the size of the raw email table is larger than 200GB. Such amounts of data cannot be processed on a standard, local machine. Thus, we used cloud technologies to process the data. The tables were queried using the Presto query engine. The query results were analyzed using various Python scripts making use of Spark (PySpark). In total, fourteen queries and 22 Python scripts were written to explore data, process data, and build models.

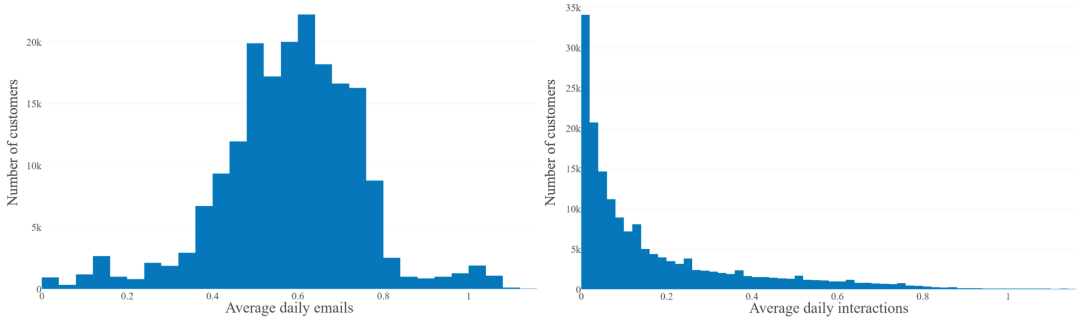


FIGURE 8.3: Distributions of emails received and emails interacted with by customers: average daily emails (left) and average daily interactions (right).

Filtering

Given the data size, data quality challenges, and to focus on relevant customers, we filter the data. In the raw data, we have approximately 240 million unique customers. However, this does not correspond to reality because business definitions, falsely identified customers, or inactive customers inflate this number. We reduce this to approximately one million customers by applying various filters. This procedure is visualized in Figure 8.4. First, we exclude stores by limiting the number of purchases and the total order value of each customer. Next, we focus only on customers that have ever placed an order, are registered (this excludes guest accounts), are flagged as customer (excludes duplicate accounts), have indicated that they can be contacted online, are situated in Europe, and have shown activity on the online platform in the previous year.

Data exploration

The retailer has over one million unique active customers in its database. In total, a little more than 132 million emails were sent, leading to around 34.5 million interactions. The main interaction category is 'email open', which occurs over five times more frequently than the second interaction category, 'link click'. This is intuitive, as an email needs to be opened in order to click a link. Even fewer emails are related to direct online sales, and rarely an email leads to an unsubscribe or deactivation (see Figure 8.1). The customers that interact with an email usually do so within a few hours. The majority even within one hour, with the number of interactions declining by the hour afterward. Only after 24 hours, there is a slight increase in the number of interacting customers (see Figure 8.2).

With the current email strategy, the retailer does not send the same emails to the same customers. The average customer receives an email every other day and interacts with an email every ten days. Interestingly, some customers interact with more than one email per day on average. The email interaction rate varies between the email category and email type. The interaction rate of individual emails shows even larger differences. This rate ranges from 3.4% to 67%. Figure 8.3 shows the average daily emails received and the average daily interactions per customer. The distributions of both statistics differ much. The average daily interactions look exponentially distributed by visual inspection, whereas the average daily emails received looks more normally distributed.

FIGURE 8.4: The various filters applied to the customer data.

In this research, we are mainly interested in delivering relevant communication to the customers. Whether an email is relevant to a customer can be expressed by whether the customer interacted with the email. We investigate two correlations related to the email interaction rate. We do this by visualizing the relation with a scatter plot (plotting a random sample of the data) and including a 95% confidence interval for the mean. The confidence interval is created through a bootstrap procedure.

Figure 8.5 (left) visualizes the correlation between the average number of emails received and the number of interactions. The average daily interactions is positively correlated with the average daily emails. This is intuitive, as it would benefit no strategy to send more emails to a customer that does not interact with emails. Also, it is impossible for a customer to interact with two emails if the customer only received one. However, sending more emails does not necessarily mean more interactions. Figure 8.5 (right) visualizes the correlation between the interaction probability and total order value of a specific customer. The interaction probability is defined as the number of interactions divided by the number of received emails for a specific customer. The graph indicates that a higher interaction probability is correlated with a higher-order value. When looking at the interaction probabilities of 0.3 and 0.4, the confidence intervals for the mean total order value (averaged over all customers) are non-overlapping. For a probability of 0.3, the confidence interval is $[174.68, 180.71]$ and for a probability of 0.4 this yields $[189.02, 195.11]$. Thus, customers that have a higher interaction probability have a higher customer value (for interaction probabilities smaller than 0.8).

8.4 Model Description

We implement a discrete-time Markov decision process (MDP) for our email marketing process. The MDP is defined by four entities: the state space $\mathcal{S}$, the action space $\mathcal{A}$, the reward function r, and the transition function p.

We define a state $s \in \mathcal{S}$ as a vector of the form $s = (x_0, x_1, x_2, y_0, y_1)$. Here, x_i represents the $(3-i)^{\text{th}}$ previous interaction of the customer for $i \in \{0, 1, 2\}$. Similarly, y_j

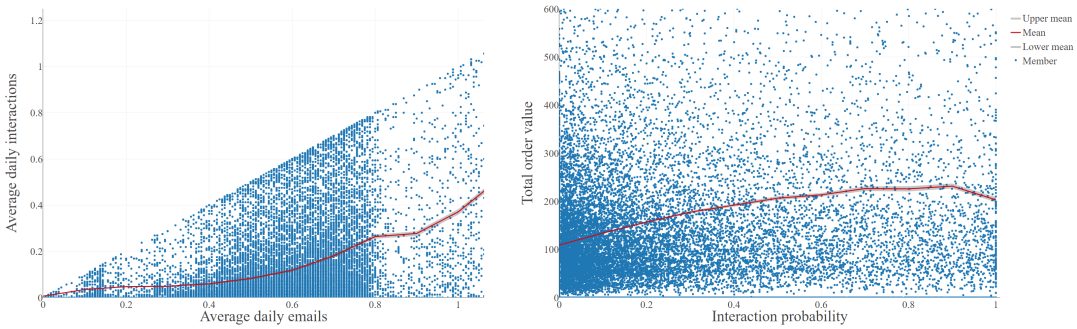


FIGURE 8.5: Scatter plots diagrams: # emails vs # interactions (left) and interaction probability vs customer order value (right).

is defined as:

$$y_j = \begin{cases} 1, & \text{if } (2-j)^{\text{th}} \text{ previous action led to an interaction,} \\ 0, & \text{otherwise,} \end{cases} \tag{8.1}$$

for $j \in \{0, 1\}$.

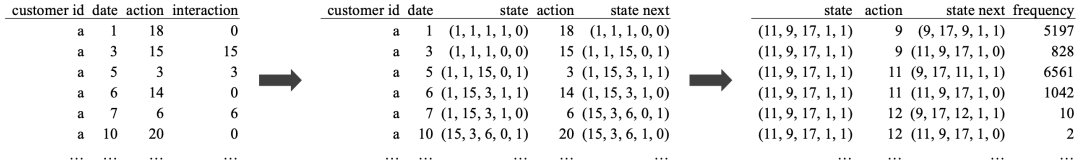
This choice for the state is partially inspired by [114], in which the state is defined as the sequence of the past k items bought. We make a clear distinction between actions and interactions, an action meaning sending an email to a customer and an interaction meaning the customer interacting with an email. The x_i's of the state space represent a customer's preference in content, and the y_i's represent the customer's sensitivity to emails. The parameters $i = 3$ and $j = 2$ have been empirically chosen, leading to an approximate model. There is a trade-off between tailoring the model for individuals and, more accurately, estimating the model parameters. The size of the state space grows exponentially as i and j are increased, since $|\mathcal{S}| = |\mathcal{A}|^i 2^j$.

We define an action $a_i \in \mathcal{A}$ as an integer. This integer represents a combination of email category and email type. An example of a category is 'household products' and an example of type is 'special event'. In our data, twenty categories and 21 types exist. However, not all combinations of category and type appear in the data. Therefore, we focus on the twenty actions that occur most frequently. In this way, we reduce the size of the action set by 95% at the cost of discarding 21% of the data.

The reward function represents the reward (business value) of a customer visiting a state. We aim to maximize the communication relevancy to the customers. This can be measured by customers interacting with emails. Thus, the reward function should measure email interactions. We define the reward function as $r(s) = y_1$ for $s = (x_0, x_1, x_2, y_0, y_1)$. This function expresses whether the previous action leads to an interaction. Conveniently, the last element in the state vector already does so.

The transition probabilities are estimated by simply counting the occurrences of a transition in the data. Specifically,

$$p(s, a, s') = \frac{C(s, a, s')}{\sum_{s' \in \mathcal{S}} C(s, a, s')},$$



customer id	date	action	interaction
a	1	18	0
a	3	15	15
a	5	3	3
a	6	14	0
a	7	6	6
a	10	20	0
…	…	…	…

customer id	date	state	action	state next
a	1	(1, 1, 1, 1, 0)	18	(1, 1, 1, 0, 0)
a	3	(1, 1, 1, 0, 0)	15	(1, 1, 15, 0, 1)
a	5	(1, 1, 15, 0, 1)	3	(1, 15, 3, 1, 1)
a	6	(1, 15, 3, 1, 1)	14	(1, 15, 3, 1, 0)
a	7	(1, 15, 3, 1, 0)	6	(15, 3, 6, 0, 1)
a	10	(15, 3, 6, 0, 1)	20	(15, 3, 6, 1, 0)
…	…	…	…	…

state	action	state next	frequency
(11, 9, 17, 1, 1)	9	(9, 17, 9, 1, 1)	5197
(11, 9, 17, 1, 1)	9	(11, 9, 17, 1, 0)	828
(11, 9, 17, 1, 1)	11	(9, 17, 11, 1, 1)	6561
(11, 9, 17, 1, 1)	11	(11, 9, 17, 1, 0)	1042
(11, 9, 17, 1, 1)	12	(9, 17, 12, 1, 1)	10
(11, 9, 17, 1, 1)	12	(11, 9, 17, 1, 0)	2
…	…	…	…

FIGURE 8.6: The three data processing steps required for estimating the transition probabilities.

in which $C(s, a, s')$ is a function that counts the number of occurrences of transitioning from state s to state s' when applying action a. To create the data to estimate these probabilities, three steps are required. First, we collect on a daily level which action and interaction was registered with which customer. Next, we compute the state of each customer based on this information. Lastly, we aggregate all state changes of all customers into one final table. These steps are visualized in Figure 8.6.

To summarize the implementation of the MDP, we present an example. This example is visualized in Figure 8.7. The example highlights that when a customer is in state $s_t = (14, 6, 10, 0, 0)$ and action $a_t = 17$ is applied, we have a 19% probability of transitioning to state $s_{t+1} = (6, 10, 17, 0, 1)$ (since $p(s_t, a_t, s_{t+1}) = p\big((14, 6, 10, 0, 0), 17, (6, 10, 17, 0, 1)\big) = 0.19$) and an 81% probability of transitioning to state $s_{t+1} = (14, 6, 10, 0, 0)$. Note that for any s_t, only two possibilities exist for s_{t+1}.

Modeling considerations

Multiple challenges arise when modeling the problem as an MDP. Most of these have been tackled by defining an appropriate MDP as done in the previous paragraphs. However, some modeling choices remain, which are described next.

8.4.1 The Unichain Condition

In order for solution techniques to work for our model, the MDP needs to be unichain. The *unichain property* states that there is at least one state $s \in \mathcal{S}$, such that there is a path from any state to s [13]. A path from z_0 to z_k of length k is defined as a sequence of states $z_0, z_1, \ldots, z_k$ with $z_i \in \mathcal{S}$ with the property that $p(z_0, z_1) \cdots p(z_{k-1}, z_k) > 0$.

The unichain property does not automatically hold when we take all states and state transitions directly from the data. This is because the chain is partially observed, so for some states, it is not observed that a specific action causes an interaction. For some states, it might only be observed that the next possible state is the current state. We solve this problem by removing all states for which fewer than two next states are observed.

8.4.2 Estimation of Transition Probabilities

In our implementation, making the MDP unichain reduces the number of observed states. A problem with the estimation of the transition probabilities is that some probabilities are based upon thousands of observations, whereas others only on a few observations. This introduces noise in the transition probabilities. To tackle this challenge, we recursively remove state transitions that occur fewer than 50 times

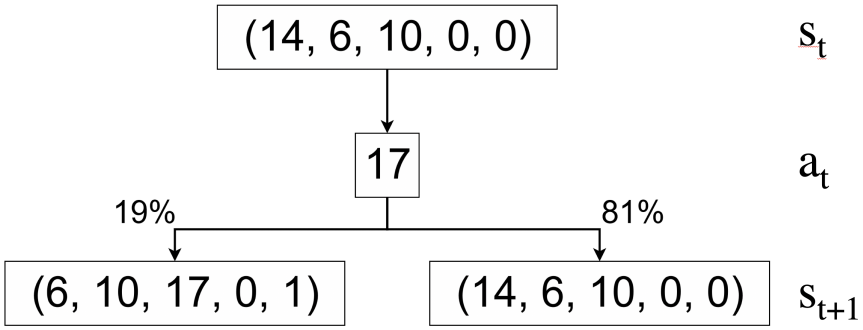


FIGURE 8.7: Example transition.

and, if this leads to states being impossible to transition to, we also remove those and transitions to those states.

The MDP is partially observed; we initially observe 86% of the theoretically possible states. After filtering, we are left with 39% of possible states. This is a large reduction in the number of observed states. However, it does ensure we focus on the most relevant and frequently observed states. Figure 8.8 shows the distribution of the number of observed transitions per state before filtering.

8.4.3 Exponential Growth

Lastly, defining and solving an MDP can be difficult because of the exponential growth of the state space due to the multiple components of the state, as discussed before, when setting the values of i and j. If the state space becomes too large, solving the MDP might not be realistic. To ensure the MDP can be solved within a feasible time period, we implement a custom version of the value iteration algorithm, taking into account the following issues.

In our case, the set of possible next states, defined as $E(s, a)$, only consists of two states. This significantly reduces the run time of the algorithm. If we did not do this, the algorithm would have to check the transition probabilities to and values of all 32,000 possible states.

We implemented the action set, $\mathcal{A}$, as being dependent on the state, thus redefining it as $\mathcal{A}(s)$. For some states, not all twenty actions are observed. So it is unknown to the model what the transitions would be. Not taking into account these unknown actions improves the performance of the algorithm.

Finally, we initialize $E(s)$, $\mathcal{A}(s)$, and $p(s, a, s')$ for all s, a, and s' in memory using Python dictionaries. This allows for $\mathcal{O}(1)$ lookup steps of any probability, action set, or the set of next states within the algorithm.

Finding the optimal policy

We find the optimal policy to the MDP by using two methods: (1) value iteration and (2) Evolutionary Computing (EC). EC can be seen as a family of algorithms that acts as a meta-heuristic. They can be applied to finding the optimal value to an MDP and potentially generating near-optimal solutions efficiently. We have not observed this way of using EC in the literature.

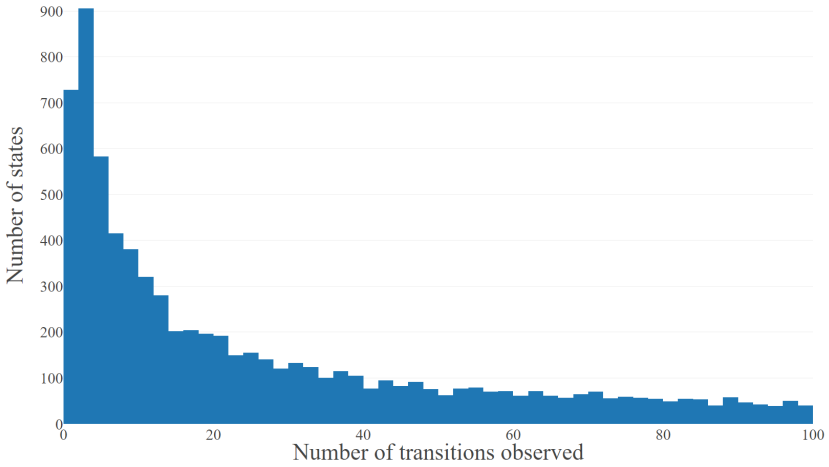


FIGURE 8.8: Distribution of the number of observed transitions per state.

Inspired by nature, EC works with notations of an individual, population, generation, selection, recombination, and mutation [**evolutionary_computing**]. A generation is a population in a certain time period, a population consists of multiple individuals, and an individual represents some solution to a problem. Individuals can recombine with other individuals to create offspring, and individuals can be mutated. Selection operators determine which individuals pass on to the next generation.

We view an individual as a strategy, thus, as a mapping from state to action. Therefore, we represent an individual as a vector of integers $\langle s_1, \ldots, s_n \rangle, n = |S|, s_i \in A$, in which s_i is a gene specifying which action to take in the state at index i in the list of possible states. We can assign a fitness f_i to individual i by applying a variation of the value iteration algorithm, in which we do not follow the optimal but the current strategy.

We initialize the population by randomly generating individuals. We choose a population size of $\mu = 100$, fitness proportionate parent selection, the uniform recombination operator, and the random reset mutation operator.

Regarding survivor selection, we use a λ-μ ratio of 2, which means we generate twice as many offspring as we have parents. We use a $(\mu + \lambda)$ survivor selection technique, we select survivors from the union of the current population and the children. The survivor selection operator we use is a tournament selection procedure of size $k = 6$. We sample k individuals from the set of parents and children, and choose the individual with the highest fitness as a survivor. We repeat this process until our new population size equals μ. Additionally, we use elitism, i.e., the best individual from the old generation is always selected for the new generation.

Our termination condition is based upon time; we stop generating offspring after running the algorithm for 24 hours.

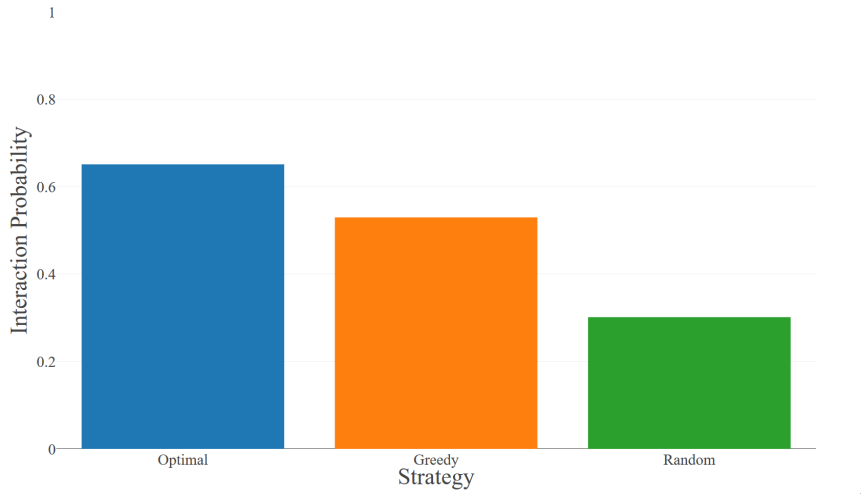


FIGURE 8.9: Comparing strategy performance: optimal vs. greedy vs. random.

Modeling Considerations

Next to the considerations of creating the MDP, two more challenges arise when modeling the EC approach. They are described as follows.

8.4.4 Choice of Components

The main challenge is choosing the components and the parameters they imply. We make these choices based upon a grid search procedure. This procedure is time-consuming, as most parameters influence the balance between exploitation and exploration, which concerns the algorithm's performance in the short- and long-term. It might seem that a parameter positively affects the fitness in the early generations. However, when looking at a longer horizon, this might not be true.

8.4.5 Evaluation of Strategies

The evaluation of strategies is time-consuming as well. To improve the performance, we use a modified version of the value iteration algorithm, in which we only follow the given strategy. This has the advantage that not every action in every state should be taken into account, and thus, the algorithm converges faster. To further reduce the evaluation time, we decrease the convergence threshold ε over the generations. In this way, the evaluation of the population is faster in the first generations and gradually slows.

8.5 Results

In this section, we present an analysis of the performance of the models. We analyze the strategy performance by comparing three different strategies, all based on the MDP framework: the optimal strategy, a greedy strategy, and a random strategy (benchmark). The optimal strategy is calculated through value iteration, the greedy strategy by choosing in each state the action with the highest interaction probability, and the random strategy by randomly choosing an action in each state.

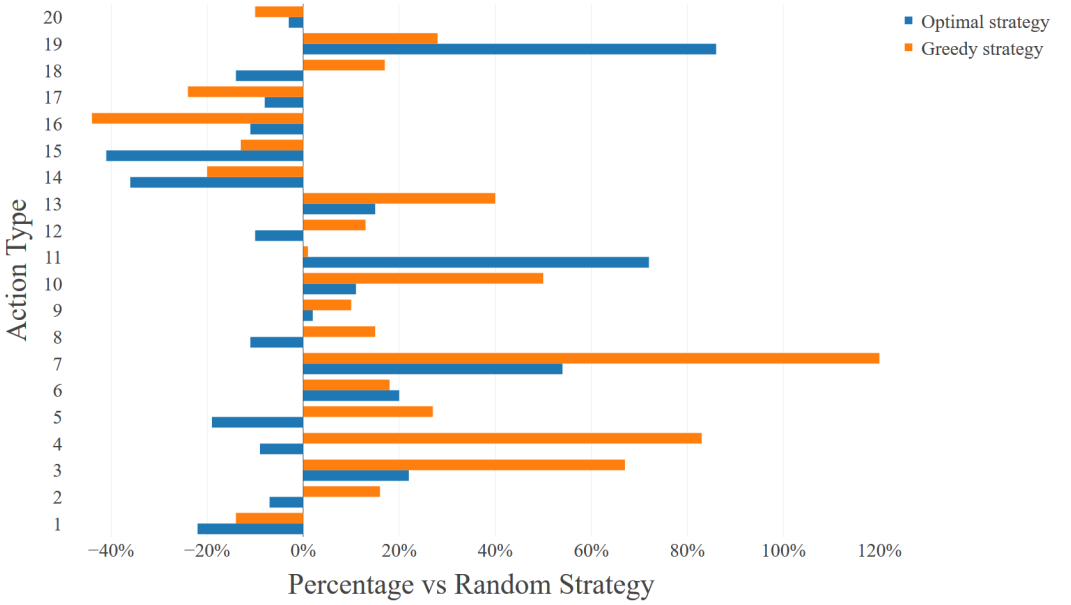


FIGURE 8.10: Action performance: frequency of an action within the optimal or greedy strategy divided by the expected frequency of that action.

Figure 8.9 shows the resulting performance of the three strategies. The optimal strategy has the highest long-run interaction probability, corresponding to a value of 65%. The greedy strategy is second with a rate of 53%, and the random strategy with 30%. Interestingly, the interaction rate of the optimal strategy is 23% higher than the rate of the greedy strategy, showing that taking into account delayed rewards can highly increase the strategy value. Both the optimal and greedy strategy perform better than the random strategy, showing that using advanced strategies has a large impact on the interaction rate.

Figure 8.10 highlights the effectiveness of each action type. This effectiveness is measured by dividing the frequency of an action within the optimal or greedy strategy over the expected frequency of that action. It is measured in this way, since an absolute measure would not be accurately representing the action performance, as in some states only one action might be possible. So the absolute measure would not represent *how much* the action is preferred over other actions. A comparison between the greedy and optimal strategy is made to highlight the difference between short- and long-term rewards of the corresponding action.

Large differences are observed in action performance. Actions that perform well on both the short- and long-term are action 7: the type retail clearance, 19: weekly product releases in a specific category, and 6: new releases. Interestingly, some actions are highly beneficial for the long-term, but not beneficial for the short-term, see., e.g., action 4. Actions that perform poorly are action 1, 14, 15, 16, or 17, which are all weekly product releases. It seems that only the weekly product release in a specific category (action 19) performs well.

Figure 8.11 shows the fitness of the best individual throughout the generations during the run of the EC algorithm. This fitness curve increases rapidly in the first generations. However, the increase slows down as the generations pass. The algorithm

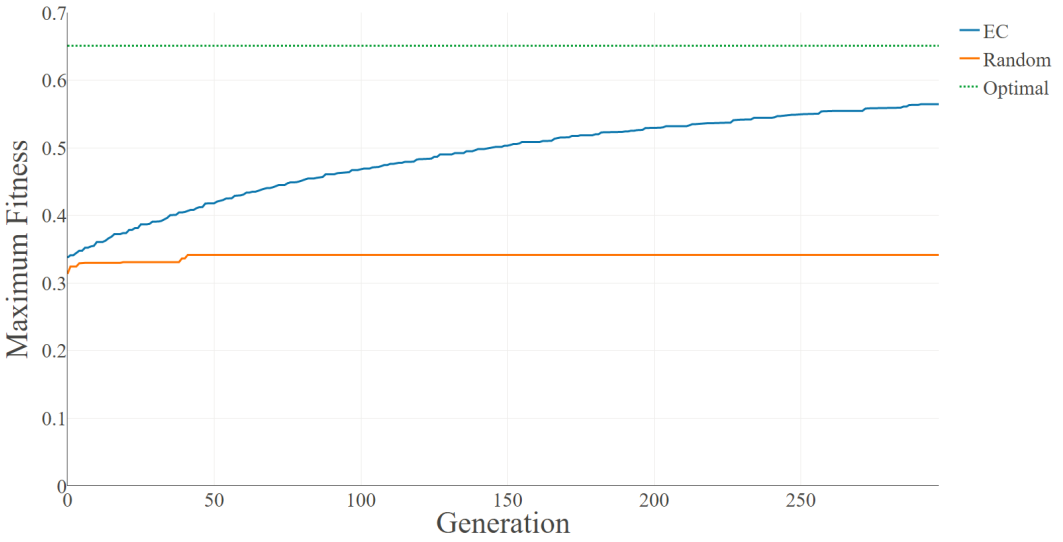


FIGURE 8.11: Performance of EC: maximum fitness per generation.

did not find the optimal strategy within the time limit of 24 hours, which corresponds to running 297 generations. Given more time, the algorithm will find better individuals and converge towards the optimal solution. The value of the optimal strategy is highlighted by the dotted line. The orange line (i.e., the line with the lowest maximum fitness) shows the maximum fitness found by a random search.

8.6 Discussion and Conclusion

This research shows that the retailer can increase its relevance to its customers by applying a different email strategy. Hereby, it possibly increases the revenue it generates. However, the strategy we developed is based on the data generated from the retailer's current email strategy. If the retailer starts experimenting with different strategies, this might uncover patterns unknown to the current model and potentially improve the optimal strategy we presented.

An interesting result of this research is the difference between the optimal and the greedy strategy. The interaction rate of the optimal strategy is 23% higher, relatively. Thus, the balance between short- and long-term rewards should be taken into account when dealing with similar problems. If we had chosen to use traditional methods, such as content-based or hybrid filtering, this result would not have been directly visible. These methods do not explicitly include this balance, so during the modeling process, it will be beneficial to try to include this balance.

Moreover, the results indicate a 'reality gap' between theory and practice. The interaction rate of the random strategy (30%) is higher than the interaction rate of the retailer's current strategy (27%). This is probably because our model has fewer restrictions compared to real life. For example, the model has a larger set of emails to choose from as in real-life some emails might be time-dependent. However, with the interaction rate of the optimal strategy being 65%, the model shows to have potential.

Throughout this research, all data concerns the past. However, to measure the impact of strategies more accurately, it would be better to measure the performance in real-time. For example, through an A/B testing procedure. Additionally, an algorithm like reinforcement learning could be used to learn the value of strategies in real-time. This algorithm is known to balance short- and long-term rewards and balance the trade-off between exploration and exploitation. It hereby tries to both learn a better strategy and apply the best-known current strategy whilst executing certain strategies.

Furthermore, we can extend the model by redefining actions. In this research, we focused on emails. However, this channel is not tied to the model. In the future, the same model can optimize push notifications of mobile applications, in exactly the same manner as the current model does.

Our research results show that EC is less efficient in finding the optimal solution than the value iteration algorithm. The value iteration algorithm converges below ϵ within twenty minutes on the same machine. Potentially, the EC approach can be improved by choosing different operators. However, the algorithm needs to be improved largely in order to match the speed of the value iteration algorithm. On our MDP, the EC approach seems inadequate; however, in other cases, it might still be a good idea to implement. For example, an MDP where the action space is larger and, therefore, the value iteration algorithm might have difficulties to converge. In this case, the EC approach can deliver better strategies than random, and if given enough time, approach the optimal solution.

Research opportunities

As with any model, the model we presented in this research is a simplification of reality. The main impact is that, compared to real life, the model can choose between more actions. In reality, not every action can be undertaken in every time period. This can be improved by further restricting the action set, based upon the state. For example, incorporating the previous action in the state and restricting the action set based on this previous action.

Furthermore, the estimate of transition probabilities can be improved. At the moment, this estimation is based upon counting frequencies. However, when transitions are not observed or observed infrequently, this estimation is unreliable and these transitions are filtered. This leads to a further restricted state space. Instead of removing these transitions, we could initialize a default probability from transitioning from a state to any other state. Or, we could use machine learning techniques to estimate these probabilities, as a transition probability might say something about the transition probability of a similar action.

9 Benefits of Social Learning in Physical Robots

9.1 Summary

Robot-to-robot learning, a specific case of social learning in robotics, enables the ability to transfer robot controllers directly from one robot to another. Previous studies showed that the exchange of controller information can increase learning speed and performance. However, most of these studies have been performed in simulation, where robots are identical. Therefore, the results do not necessarily transfer to a real environment, where each robot is unique per definition due to the random differences in hardware. In this chapter, we investigate the effect of exchanging controller information, on top of individual learning, in a group of Thymio II robots for two tasks: *obstacle avoidance* and *foraging*. The controllers of the robots are neural networks that evolve using a modified version of the state-of-the-art NEAT algorithm, called *cNEAT*, which allows the conversion of innovations numbers from other robots. This chapter shows that robot-to-robot learning seems to at least parallelise the search, reducing wall clock time. Additionally, controllers are less complex, resulting in a smaller search space.

9.2 Introduction

To enable autonomous robots to operate reliably in an environment that is not fully understood or known at design time, there is a need for self-learning robots. In such a setting, the robots can learn individually, e.g., by encapsulating a self-sufficient learning algorithm within the robot. These robots learn about, and act, in their environment while completing a task.

The robotic controller that we consider for the robot to learn a task is a neural network. This is a direct policy that maps the sensor inputs of the robot to actions. This mapping, consisting of nodes and connections between the nodes, are evolved with evolutionary algorithms.

Evolutionary algorithms are inspired by Darwins' theory of survival of the fittest. In nature, animals survive and procreate when they are more fit. Similarly, a robotic controller is tested by observing the behaviour of the robot and is given a corresponding fitness measure. The higher the fitness, the higher chance this controller has to procreate. Over generations, the quality of the controllers will improve and lead to robots that are capable of executing a predefined task properly.

The robotic controllers are evolved online, while the robot is performing the task, as opposed to offline learning, where only the best controller is transferred to hardware. Online learning increases the difficulty of the learning process for two reasons. First, if the robot is stuck in a difficult situation, e.g. a room with one small opening,

the first and only priority is to recover from this situation. With offline learning, a robot will be repositioned at the end of the controller evaluation. Second, it is important that the whole population of controllers is performing well because the robot is already performing the task. Therefore, the measurements of the performance in this chapter always include all individuals in the population.

The choice of online learning is perpendicular to the choice for a physical or simulated platform. In this chapter we choose a physical platform. Online learning on a physical platform has the disadvantage to be slow.

Accelerating the learning process could be reached when a collective of autonomous robots is used that can share knowledge, i.e. *socially learn*, to enhance the individual learning process. Note that social learning in robotics is not the same as the widely used definition of social learning: learning through observation of conspecifics. Regarding robots, we can add a type of social learning, that we call robot-to-robot learning, based on the ability to transfer robot controllers directly from one robot to another. (In common parlance, this would be the robotic equivalent of "telepathy".)

The benefits of robot-to-robot learning have been shown in multiple simulation studies [43, 139, 69]. In particular, evidence by [60] and [118] suggests that robot-to-robot learning can linearly decrease learning time in terms of number of robots, i.e., the fitness measure that four robots can reach in two hours can be reached by eight robots in one hour when they learn socially. Evidence by [52] shows that for a range of parameter settings, learning speed usually increases when applying robot-to-robot learning. The learning speed increase is lower for the better parameter setting, which are parameter settings that result in a higher performance for the individual learning robot.

Implementations of a physical robotic collective learning a task are rare. The authors of [53] implemented an obstacle avoidance task and [94, 119] looked at the phototaxis task. These tasks are simple tasks that do not require the robot to have a camera. The authors of [55] did use a camera for the relative complex foraging task. This task demands from the robot to collect pucks to bring to a designated target area. However, the controller evaluation time in this implementation was too short (under four seconds) to score a goal, which resulted in the necessity to keep a variety of behaviours in the population. Additionally, the robotic group size did not vary in these studies. Therefore, no systematic study has been performed on the benefits of robot-to-robot learning in a physical robotic group for multiple tasks and different group sizes.

The main objective of this chapter is to examine the increase in learning speed and performance due to robot-to-robot learning, on top of individual learning, for varying robotic group sizes. Additionally, we hypothesize that robot-to-robot learning will lead to more robust solutions, as each robot has small random differences. That is to say, some cameras might be mounted under a marginally different angle for example. Learning correct behaviour in experiments with more than one robot means that optimal performance is generalised behaviour over an entire group. Intuitively, more robust solutions are those that are less complex.

Learning is implemented by evolving neural networks with a state-of-the-art evolutionary algorithm called NeuroEvolution of Augmenting Topologies (NEAT). NEAT was designed as a general method that can be applied to any (robotic) task, and for this study, we chose two tasks: (1) obstacle avoidance and (2) foraging. To observe

the increase in learning speed and performance due to robot-to-robot learning, we compare one learning robot with a group of four and eight robots.

The robots operate in their own arena. Consequently, the performance of the robot is only due to its own actions and not influenced by other robots in the same arena. However, the robots do communicate across arenas. Removing this inter-robot collision allows for a better comparison between the individuals and the robot-to-robot learning experiments.

The NEAT algorithm is not directly applicable for robots that exchange knowledge. We, therefore, propose an extension of NEAT, named *cNEAT*. This extension allows the conversion of innovation numbers from other robots.

This chapter shows promising results when applying robot-to-robot learning. We show that on top of a parallelisation of the search, robots that learn socially are more capable of retaining the best controllers. Additionally, the controllers are less complex resulting in a smaller search space, which is extremely beneficial when using physical robots. It is shown that a relatively complex task, such as the foraging, can be learned within the hour. Therefore, online learning methods can be tested on real robots for more complex tasks than currently possible.

9.3 Learning Mechanisms

9.3.1 Individual Learning Mechanism

Individual learning takes place through an encapsulating, self-sufficient learning mechanism. The learning mechanism used in this chapter is NEAT [133]. NEAT is a state-of-the-art evolutionary algorithm that evolves both the topology and the connectivity of artificial neural networks. In NEAT, an initial population of neural networks without hidden layer is randomly generated. These networks will be referred to as *individuals* in the population.

Over generations, nodes and connections can be added to individuals. In order to compare different individuals, the changes are stored as innovation numbers. In the original implementation of NEAT, innovation numbers are only used within one generation. As a result, identical innovations could have different innovation numbers when they occur in different generations. Therefore, networks that are similar could have a larger distance measure and could be placed in different species. Therefore, we propose to keep the innovation numbers over generations.

9.3.2 Robot-to-Robot Learning Mechanism

Next to the individual learning mechanism, a robot-to-robot learning mechanism takes place as well. First, for each robot, all the individuals in the population are evaluated. Thereafter, the robots exchange information. Each robot sends its best controller to the others and from all received controllers, each robot chooses one controller based on so-called fitness proportionate selection. The robot adds this controller to the population, whereafter new offspring is created.

As noted before, NEAT can modify the topology of the neural networks during evolution. Every structural modification in the network is identified by a unique innovation number to enable alignment of genomes for recombination purposes. When implementing NEAT with the possibility to exchange individuals as described for

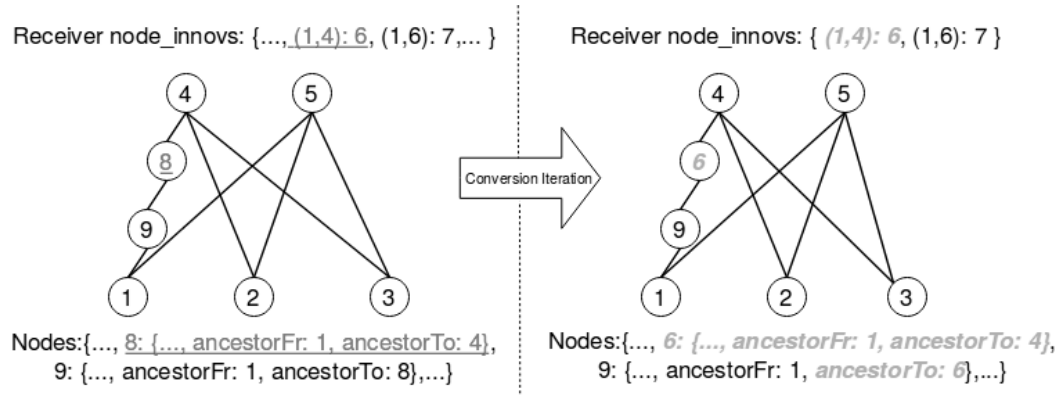


FIGURE 9.1: A first iteration of the conversion algorithm with two conflicting node IDs.

robot-to-robot learning, care must be taken to avoid conflicting innovation numbers. Previous work [120] solved this by using timestamps as innovation numbers. However, using timestamps results in a unique innovation for every mutation. This results in the same problem as described before, where two similar networks result in a larger distance than they actually might have. Thus, we propose an adjusted approach called *conversion NEAT*, shortly *cNEAT*.

9.3.3 cNEAT

When robot 1, R1, receives a network from robot 2, R2, R1 needs to match the node IDs of the received network from R2 with that of its own. This is because R2 might have assigned the same innovation ID number to a different innovation as opposed to R1.

To solve this problem, R1 iterates over the nodes and converts the node IDs accordingly. This ensures that:

1. the nodes of the received network from R2 that match a node innovation of R1 have the same ID;
2. the nodes that do not match with any innovation of R1 get a new ID that is not assigned to any node of R1. This new innovation ID is thereafter added to the list of node innovations of R1.

To better understand the conversion, we provide an example shown in Figure 9.1, on the left. The node IDs of the receiving robot R1 are on top and the nodes of the received network from R2 are on the bottom. The received network has two conflicting IDs. Node ID 8 and 9 are placed between nodes 1 and 4. However, the node innovations list of R1 claims that the node in that position should have ID 6 and 7.

Node 9 is in a more difficult situation than node 8: the ancestors node (i.e., ancestorFr and ancestorTo) of node 9 do not match with the ones specified in the node innovation list, due to the wrong assignment of node 8, that should be node 6. Thus, we first need to convert node 8 into node 6 (shown in Figure 9.1, on the right), and only then we will be able to match and convert node 9 into node 7.

It is important to note that the ancestors' information of a node (i.e., ancestorFr and ancestorTo) could help to understand the order of creation of the nodes. In fact,

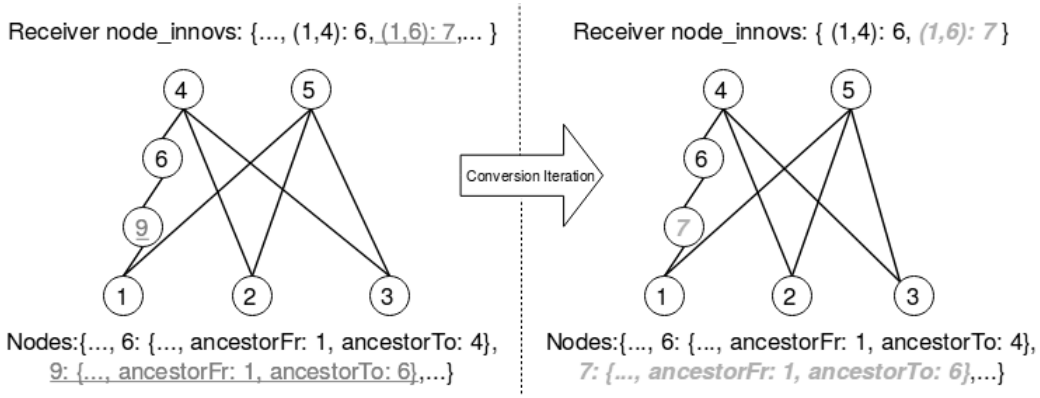


FIGURE 9.2: The second and last iteration of the conversion algorithm. The received network has one conflicting ID left (9 should be 7).

nodes 8 and 9 are both placed between nodes 1 and 4. However, node 8 was first created (its ancestors are 1 and 4), and only during a next mutation node 9 was created (its ancestors are 1 and 8). For that reason, the conversion of node 9 depends on the conversion of node 8.

As a consequence, a conversion could trigger other conversions, because some wrong IDs could avoid the matching between the ancestors of the node and the node innovations list. As a result, after the first conversion shown in Figure 9.1, we can apply a second conversion, shown in Figure 9.2.

The conversion algorithm needs to iterate over the node ID numbers several times until all the conversion is performed. Within each iteration, the algorithm matches the information about the position of a node (ancestors) and its ID with the node innovation of the receiving robot (in our example R1): if the ID is wrong, the information of the node and every reference to it (e.g., ancestor information in other nodes with references to that node) are converted to the ID used in R1.

Algorithm 4 summarises the individual and robot-to-robot learning mechanism in pseudocode. The NEAT algorithm is a framework where the specific implementation for the variation and selection operators are not set. In the experimental setup, the list of used parameters is provided to clarify the chosen mechanisms that we used for our specific implementation.

9.4 Tasks

The learning mechanism is deployed on Thymio II robots to learn two tasks: obstacle avoidance and foraging. The foraging tasks require the robot to collect pucks to bring to a target area. We extend the standard Thymio set-up with a more powerful logic board, a camera (only used for the foraging task), wireless communication, and a high capacity battery. We use a Raspberry Pi 3 that connects to the Thymio's sensors and actuators and processes the data from the Raspberry Pi Camera. The WiFi is integrated with the Raspberry Pi and enables inter-robot communication.

Algorithm 4 Pseudocode of the algorithm that runs on each robot

```
initialise population of first generation (P_1) with individuals i_1, ..., i_n
while current generation ≤ final generation do
  for i in P do
    evaluate i
    store fitness of i
  end for
  sort the individuals based on fitness (i_1 is best)
  if robot-to-robot learning then
    send i_1 to all other agents
    receive best individual from all other agents
    pick one individual r_1 using fitness proportionate selection
  end if
  generate offspring by:
      adjust specie fitness based on age
      pick parent pool per specie
      calculate number of specie offspring with roulette wheel selection
      clone specie best
      pick parents based on tournament selection
      apply mutation or crossover and mutation
      add offspring to specie
end while
```

9.4.1 Obstacle Avoidance

The obstacle avoidance task requires the robot to drive as fast as possible through an environment without hitting the walls. The network inputs are the seven proximity sensors around the robot and the outputs are the motor speeds for the left and right wheel. Including the bias node, this results in an initial network of sixteen weights. The Thymio II robot in an empty environment for the obstacle avoidance task is shown in Figure 9.3.

There is a common fitness function to evaluate the robots' performance for the obstacle avoidance task. Given an evaluation period of T time steps, this is measured as follows:

$$f = \sum_{t=0}^{T} s_{trans} \times (1 - s_{rot}) \times (1 - v_{sens}), \tag{9.1}$$

where:

- s_{trans} is the translational speed, calculated as the sum of the speeds assigned to the left and right motor and normalised between 0 and 1;
- s_{rot} is the rotational speed, calculated as the absolute difference between the speed values assigned to the two motors and normalised between 0 and 1;
- v_{sens} is the value of the proximity sensor closest to an obstacle normalised between 0 and 1.

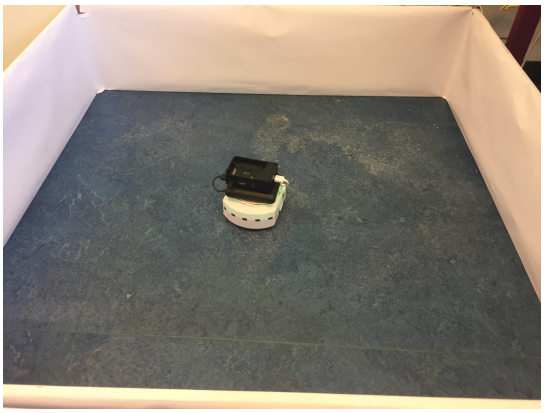


FIGURE 9.3: The environment with one robot for the obstacle avoidance task. This setup is duplicated for the number of robots used in the experiment.

9.4.2 Foraging

A foraging task requires the robot to collect pucks and bring them to the nest located in a corner of the arena. The extended Thymio II and the environment is shown in Figure 9.4.

We use the camera image to define three task-specific sensors: puck in sight, puck in gripper and goal in sight. The goal is visible even if the robot does not have a puck yet. This makes the task more complex. Additionally, we use the sum of proximity sensors in the front and two proximity sensors in the back to determine if the robot is colliding with the wall. As a result, the neural network controller has six inputs, including bias. The number of output nodes is again two, where each node corresponds to a wheel of the robot, indicating the speed. This results in an initial network size of fourteen weights.

The *fitness* of a robot is defined as:

$$f = c_1 \times n_{walls} + c_2 \cdot n_{puck} + c_3 \cdot n_{goal}, \tag{9.2}$$

where:

- n_{walls} is the number of time steps the front and back proximity sensors are not activated, meaning no wall is being hit;
- n_{puck} is the number of pucks collected during an evaluation;
- n_{goal} is the number of goals scored during an evaluation.

The hyperparameters c_1, c_2 and c_3 are empirically put at 1, 1,000 and 10,000 respectively.

With online learning, the next individual inherits the state of the current individual automatically. When the controller evaluation ends with a puck in the gripper, the next individual starts with this puck without any effort. For this reason, a puck only count as collected if the robot did not have a puck within its gripper before. This is done by subtracting the fitness of the sum of the three time steps before to the fitness at the current time step. This means that when a goal is actually scored, the robot has a fitness of around 7,000. If the fitness of the time step results in a negative value, the fitness value is set to 0.

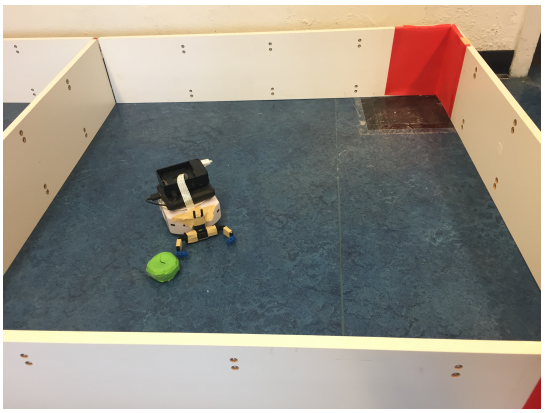


FIGURE 9.4: The environment with one robot searching for the green puck to bring to the red target location. This setup is duplicated for the number of robots used in the experiment.

9.5 Experimental Setup

We distinguish two different sets of experiments: (1) the *obstacle avoidance* task and (2) the *foraging* task. For each setup, we compare individual learning only and individual and robot-to-robot learning together. The learning of the robot is conducted online, i.e. the robot is not relocated between the evaluations and each controller is tested starting from the location reached by the previous one.

For the foraging task, there is a necessity to calibrate the cameras of the robots automatically before the start of every run, because the orientation of the cameras across robots can be different. This orientation means that the size of the puck in each of the frames that the robot shoots can vary. After the calibration, a hand-coded controller is used to test the calibration process; the hand-coded controller should exhibit the optimal behaviour: turning until the puck is in sight, drive straight until the puck is in the gripper, turn until the goal is in sight, and drive straight to the goal. Because the quality of the camera is not what we want it to be, we added Velcro on the gripper and the puck. As a result, once the robot has the puck, the corresponding sensor input of having the puck will be set to true until a goal is scored (even if the robot does lose the puck). We choose this implementation, because the camera is not always accurate enough to see the colours appropriately.

Human intervention is necessary when the robot is in the corner facing the wall and tries to turn right against the wall and the motor of the robot is not powerful enough to push itself back. When the robot collects a puck for the foraging task, we relocate the puck to a random position in the arena and place the robot just behind the black square.

Robot-to-robot learning experiments are performed with a group of four and eight robots. A populations size of 24 is used for the individual learning experiments resulting in a population size of six and three for the four and eight robot setup respectively. The number of generations is nineteen, restricted by battery capacity for the one robot experiment, and one evaluation is twenty seconds for the obstacle avoidance task and 60 seconds for the foraging task. This results in a time required per experiment of the obstacle avoidance task of around 160 minutes, 40 minutes and twenty minutes for one, four, and and robots. And for the foraging task of eight

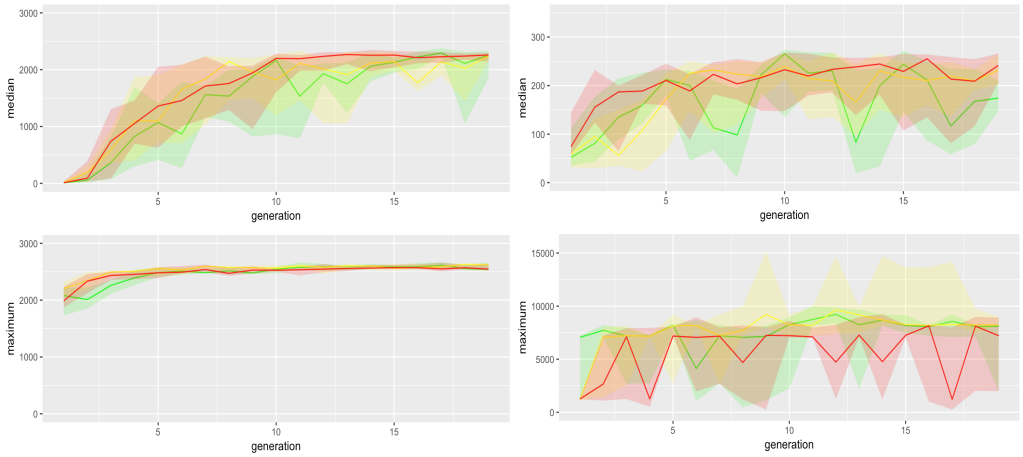


FIGURE 9.5: Median (top) and maximum (bottom) performance with interquartile range over generations for the obstacle avoidance (left) and foraging task (right). The individual learning robot is presented by the green colour. Robot-to-robot learning with four robots is presented in yellow and with eight robots in red. The results are compiled over ten replicate runs.

hours, two hours and one hour for one, four, and eight robots. For all experiments, ten replicate runs are performed with different random seeds.

The code for the implementation is available on the first author's website. The most important parameter settings for the NEAT algorithm are presented in Table 9.1.

9.6 Experimental Results

In this section, the experimental results of the tasks reported in Section 9.4 using the setup described in Section 9.5 are discussed. Due to the small number of runs, some explanations are qualitative rather than quantitative.

9.6.1 Performance

In Figure 9.5, we compare the median and maximum fitness, including interquartile range, over generations for the obstacle avoidance task (left) and the foraging task (right). The individual learning robot is presented by the green colour. Robot-to-robot learning with four robots is presented in yellow and with eight robots in red. From these graphs, we can draw several conclusions.

First, for both tasks the performance improves over time, meaning that the robots learn over time. For the obstacle avoidance task, the median performance is much closer to the maximum performance than for the foraging task. This indicates that the foraging task is more difficult to learn for the robot. Second, adding robot-to-robot learning, results in a more robust median performance, showed by a smoother curve. Although the median performance for the foraging task seems to increase when using more robots, the maximum fitness seems to be lower when using eight robots when looking at the interquartile range.

Figure 9.6 shows a more detailed view of the performance over time for the obstacle avoidance task (top) and foraging task (bottom). It specifically shows the fitness

TABLE 9.1: System parameters, descriptions and used values.

Mutation and crossover parameters	
$\mathbf{p_{Xover}}$ chance to apply crossover	**0.75**
$\mathbf{p_{Mutation}}$ chance to apply only mutation	0.25
$\mathbf{p_{weightMutation}}$ chance to apply mutation on weight	**0.4**
$\mathbf{p_{WeightReplaced}}$ chance to replace weight	**0.05**
$\mathbf{p_{Connection}}$ chance to enable / re enable a connection	**0.01**
$\mathbf{max_{Perturb}}$ maximum allowed change on weight	**0.75**
$\mathbf{p_{AddLink}}$ chance to add a link	**0.1**
$\mathbf{p_{AddNode}}$ chance to add a node	**0.05**
Species parameters	
$\mathbf{species_{Target}}$ number of target species.	**2**
$\mathbf{coeff_{Excess}}$ used for species compatibility score	**1**
$\mathbf{coeff_{Disjoint}}$ used for species compatibility score	**1**
$\mathbf{coeff_{Weight}}$ used for species compatibility score	**0.7**
threshold used for species compatibility score	**2**
$\mathbf{threshold_{Change}}$ used to change threshold value	0.1
$\mathbf{species_{AgeThreshold}}$ age to count as old	8
$\mathbf{species_{YouthThreshold}}$ age to count as young	3
$\mathbf{age_{Penalty}}$ fitness multiplier for old individual	**0.5**
$\mathbf{age_{Boost}}$ fitness multiplier for young individual	**1.2**
Other parameters	
size population size of all robots combined	**24**
$\mathbf{survival_{Threshold}}$ top % individuals that can be parents	0.6
$\mathbf{tournament_{size}}$ size of tournament to select parent	**2**
$\mathbf{copy_{Best}}$ clone best specie individual previous generation	**TRUE**
$\mathbf{copy_{BestEver}}$ clone best individual so far	**FALSE**

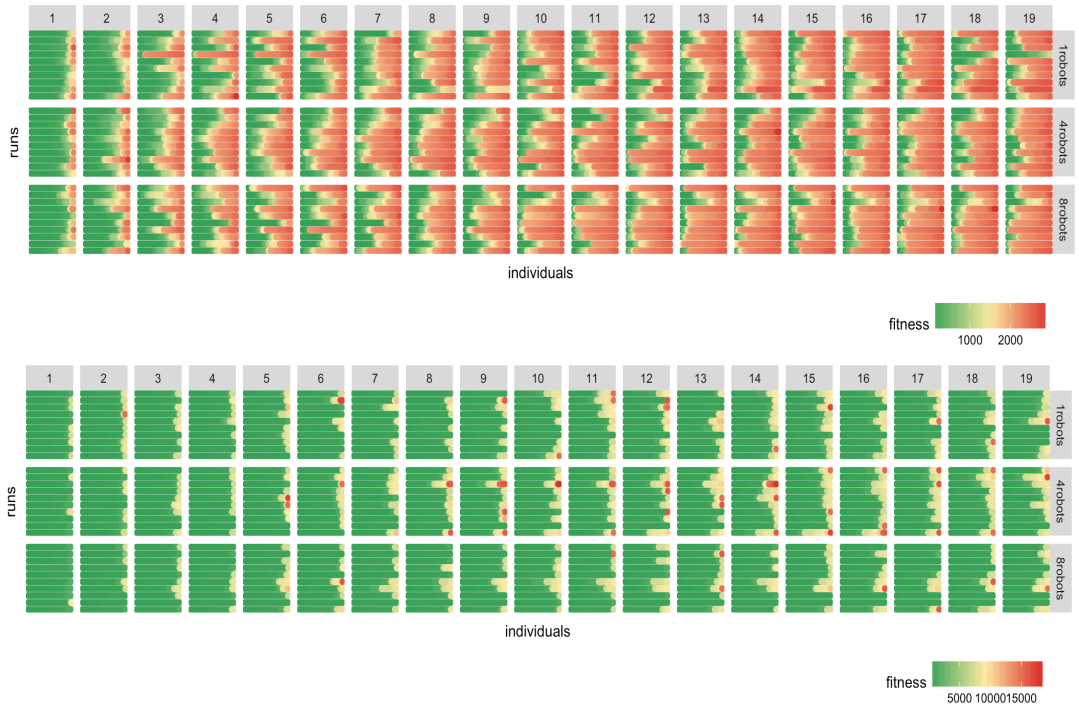


FIGURE 9.6: Fitness of the obstacle avoidance task (top) and foraging task (bottom). The x-axis presents the individuals per generation. The y-axis presents the run. The rows represents the results for one, four, and eight robots. Within one run, the individuals are sorted on fitness of which the colour reflects the value. When using multiple robots, the individuals of the final generation for all robots are combined and sorted on fitness.

of the individuals over the generations for all ten independent runs and all robot-to-robot learning experiments. To explain this graph, pick one bar where the colour goes from green to red. This bar consists of dots, where each dot represents one individual in the generation presented at the top of the column and the colour represents the fitness of the individual. There are groups of ten bars where each bar represents one run. This block of ten runs is shown for every generation, shown at the top of the column, and the number of robots, shown at the right of the row. When using multiple robots, the individuals of the final generation for all robots are combined and sorted by fitness.

We can see that for the obstacle avoidance task, there are many well-performing controllers, while for the foraging task the good controllers are sparse. However, the effect of robot-to-robot learning seems to be similar: when a high performing controller is found, there is more chance that the next generation has a high performing controller too. This results in the smoother median curve of Figure 9.5. The difference in maximum performance for the foraging task shown in 9.5 can also be better understood with this graph. It seems that a group of four robots is more capable of keeping the good knowledge in the population than using a group size of 8. This might be due to the decrease in population size from six to an even smaller size of three when using four and respectively eight robots.

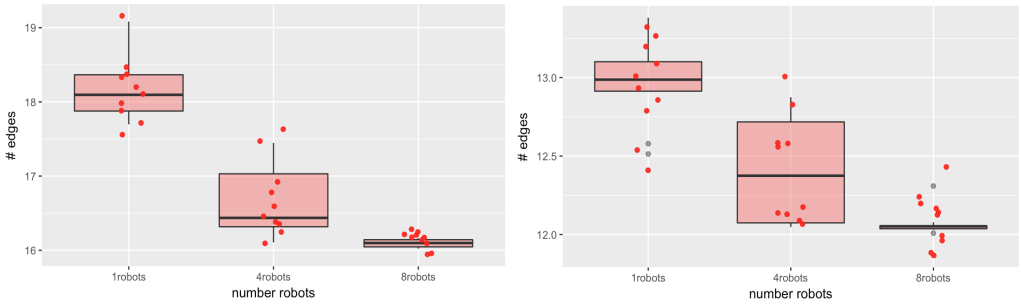


FIGURE 9.7: Median network complexity with interquartile range at the final generation for the obstacle avoidance task (left) and foraging task (right). Network complexity is expressed as the average number of edges in the network. The results are compiled over ten replicate runs.

9.6.2 Network Complexity

In Figure 9.7, we can see the median and interquartile range of the average number of edges in the final generation over the ten replicate runs for the obstacle avoidance task (left) and foraging task (right). The network complexity is expressed as the number of edges, as the increase in the number of nodes already implies the increase in the number of edges. We can confirm our hypothesis that the complexity of the network significantly drops when the robotic group size increases, indicated by the non-overlapping quartile range [95]. However, we must note that when using more robots, the chances of creating extra nodes and edges go down because a higher percentage of the population is used by the clone component of the algorithm.

9.6.3 Selection Pressure

Although the algorithm that is running on each robot is the same and the same parameters are used, the overall dynamics of the system are different. An analysis of the selection pressure will show this. The selection pressure expresses the correlation between the fitness of an individual and the number of offspring. We specifically use the Kendall's τ-b measure [49]. In short, it measures the correlation between fitness and number of offspring. A high value for τ-b indicates a strong correlation between fitness and offspring and therefore high selection pressure.

In Figure 9.8, the selection pressure over generations are shown for one (green), four (orange), and eight (red) robots. On the left, the selection pressure is aggregated over all robots, while on the right the selection pressure for the first robot is isolated, meaning only one robot is chosen in the robot-to-robot learning experiments to express the selection pressure.

Looking at the left graph, we can observe that the selection pressure for one robot is a bit higher than for more robots. This is logical because the whole population is divided over multiple robots and only the population of one robot is considered to create offspring. While the overall best controller can be selected every time for the one robot experiment, it does not participate in the creation of offspring on the other robots. As a result, the correlation between fitness and number of offspring is not as high as the correlation of one robot.

The right graph shows the selection pressure on one robot (robot number one). Therefore, the selection pressure for the one robot experiments is identical to the left

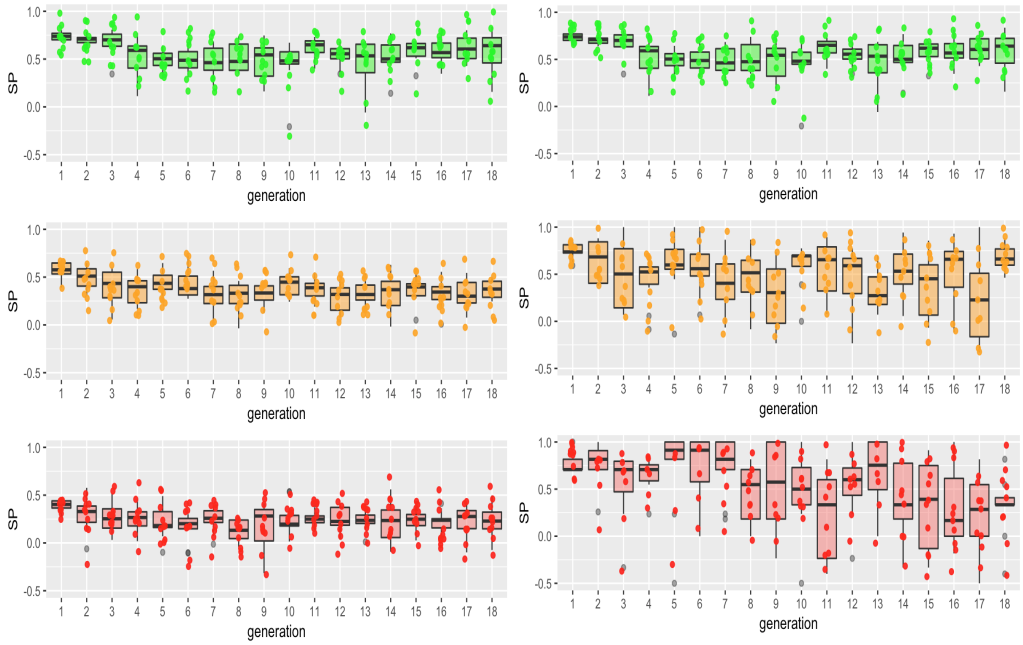


FIGURE 9.8: Selection pressure over generations for one robot (green), four robots (orange) and eight robots (red), including interquartile range over generations for the foraging task combined over all robots (left) and robot number one (right), meaning that the results of only one robot are presented for the robot-to-robot learning experiments. The results are compiled over ten replicate runs.

graph. For the robot-to-robot experiments, we see an increase in the selection pressure but also an increase in volatility. This is because, on one robot, the population size decreases and the best controller has more impact on the creation of offspring when using tournament selection.

9.7 Discussion and Conclusion

In this chapter, the effect of robot-to-robot learning on the learning speed, performance and network complexity using physical robots is investigated for two tasks.

The results show that the median performance is more robust when using more robots. Although the increase in robustness is not overwhelming, robot-to-robot learning at least distributes the search and reduces the wall clock time needing to learn a specific task without loss of performance compared to an individual learning robot.

The increase in robustness of the median performance is probably due to the fact that more robots are more likely to retain the controllers of the good performing controllers. A possible explanation for this is that good controllers are tested by multiple other robots that all start at a different position. Even though the fitness function of a foraging task is very stochastic, a good controller will probably perform well on one of the other robots.

Additionally, we have shown that using more robots results in less complex controllers. While this might be a logical result of our specific implementation details, the fact remains that we can reach a similar performance level with less complex

controllers when using multiple robots. A decrease in controller complexity means that the search space is reduced: this is especially useful in a physical setup where the time is an important restriction.

Observing the selection pressure showed that the overall dynamics change when using multiple robots, even though the algorithm on the specific robot remains identical. We showed that the selection pressure over all robots decreases while the selection pressure on the individual robot increases. This might result in an overall better exploration versus exploitation balance.

One can argue that our specific implementation of robot-to-robot learning has a link with parallel EAs and island models [7, 148]. Most of the work in parallel EAs and island models are focussed mainly on runtime analyses [7]. Measuring this in ER is of lower importance because the evaluation time of the robot is much larger than the computational effort. Additionally, the fitness function in evolutionary robotics is extremely stochastic. This is due to the specific location of the robot and the behaviour required in that location. This relevance of the location of the robot is not present with parallel EAs and island models. Despite the differences, we do believe that there are some common elements too. Especially, studying the effect of the number of islands/robots on the diversity of the whole population is of interest to both fields.

It is clear that using multiple robots changes the dynamics of the learning algorithm. However, based on the limited number of physical experiments executed in this chapter, we are unable to identify the explicit "magic" of robot-to-robot learning. In future work, we will return to a simulation platform for an in-depth analysis of the impact of robot-to-robot learning. Because the robots learn in an online fashion, the learning mechanisms in the physical robots and the simulation platform will be identical. Therefore, we can use results in simulation to validate the physical experiments.

To conclude, this chapter showed some promising results when applying robot-to-robot learning. It is shown that a complex task, such as the foraging, can be learned within the hour due to robot-to-robot learning. Therefore, online learning methods can be tested for more complex tasks than currently possible when using robot-to-robot learning. This will help the field of ER to be a powerful alternative to hand-coded robots for an environment not fully known to the designers.

9.8 Acknowledgments

This work is supported by the European Union's Horizon 2020 research and innovation programme under grand agreement No 640891 (DREAM project). A special thanks to Thomas Webbers for his support with the experiments.

Bibliography

[1] Muhammad Abdullah et al. "Exploring the Impacts of COVID-19 on Travel Behavior and Mode Preferences". In: *Transportation Research Interdisciplinary Perspectives* 8 (2020), p. 100255. ISSN: 2590-1982.

[2] G. Adomavicius and A. Tuzhilin. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". In: *IEEE Transactions on Knowledge and Data Engineering* 17.6 (2005).

[3] G. Adomavicius et al. "Context-Aware Recommender Systems". In: *AI Magazine* (2011).

[4] AeroDataBox. *AeroDataBox API*. `https://www.aerodatabox.com/`. [Online; accessed 17-06-2021]. 2019.

[5] Mayur Agarwal. *What is The Speed of a Ship at Sea?* 2020. URL: `https://www.marineinsight.com/guidelines/speed-of-a-ship-at-sea/`.

[6] AIS Hub Exchange. *AIS Data Sharing and Vessel Tracking by AISHub*. 2021. URL: `https://www.aishub.net/`.

[7] Enrique Alba and Marco Tomassini. "Parallelism and Evolutionary Algorithms". In: *IEEE Transactions on Evolutionary Computation* 6.5 (2002), pp. 443–462.

[8] R. Aroussi. *yfinance*. Version 0.1.55. Sept. 2020. URL: `https://github.com/ranaroussi/yfinance`.

[9] Charles Arthur. "Tech Giants may be Huge, but Nothing Matches Big Data". In: *The Guardian* (Aug. 23, 2013). URL: `https://www.theguardian.com/technology/2013/aug/23/tech-giants-data`.

[10] Diego Maria Barbieri et al. "Impact of COVID-19 Pandemic on Mobility in Ten Countries and Associated Perceived Risk for all Transport Modes". In: *Plos One* 16.2 (2021), e0245886.

[11] I. Benenson et al. "Public Transport Versus Private Car GIS-Based Estimation of Accessibility Applied to the Tel Aviv Metropolitan Area". In: *The Annals of Regional Science* 47.3 (2011), pp. 499–515.

[12] L. Bertolini, F. Le Clercq, and L. Kapoen. "Sustainable accessibility: a conceptual framework to integrate transport and land use plan-making. Two test-applications in the Netherlands and a reflection on the way forward". In: *Transport policy* 12.3 (2005), pp. 207–220.

[13] S. Bhulai and G. Koole. *Stochastic Optimization*. 2014. URL: `https://obp.math.vu.nl/edu/so/notes.pdf`.

[14] M. Bierlaire. *PandasBiogeme: a Short Introduction*. Tech. rep. TRANSP-OR 181219. 2018.

[15] Ane Blázquez-García et al. *A Review on Outlier/Anomaly Detection in Time Series Data*. 2020. arXiv: `2002.04236 [cs.LG]`.

[16] S. Bouton et al. *Urban Mobility at a Tipping Point*. Tech. rep. 2017. URL: https://www.mckinsey.com/business-functions/sustainability/our-insights/urban-mobility-at-a-tipping-point.

[17] Marc Brysbaert. "How Many Words do we Read per Minute? A Review and Meta-analysis of Reading Rate". In: *Journal of Memory and Language* 109 (2019), p. 104047. ISSN: 0749-596X. DOI: https://doi.org/10.1016/j.jml.2019.104047. URL: https://www.sciencedirect.com/science/article/pii/S0749596X19300786.

[18] Evgeny Burnaev and Vladislav Ishimtsev. "Conformalized Density- and Distance-Based Anomaly Detection in Time-Series Data". In: *arXiv e-prints* (2016). eprint: 1608.04585.

[19] M. Cappellari et al. "The ATLAS^{3D} project - XX. Mass-size and mass-σ Distributions of Early-Type Galaxies: Bulge Fraction Drives Kinematics, Mass-to-Light Ratio, Molecular Gas Fraction and Stellar Initial Mass Function". In: *MNRAS* 432 (2013), pp. 1862–1893. DOI: 10.1093/mnras/stt644. eprint: 1208.3523.

[20] Charles Carlstrom and Timothy Fuerst. "Monetary Policy and Self-Fulfilling Expectations: the Danger of Forecasts". In: *Economic Review* Q I (2001), pp. 9–19.

[21] Kevin M. Carter and William W. Streilein. "Probabilistic Reasoning for Streaming Anomaly Detection". In: *2012 IEEE Statistical Signal Processing Workshop (SSP)* (2012). DOI: 10.1109/SSP.2012.6319708.

[22] *CBS*. 2020. URL: https://www.cbs.nl/nl-nl/dossier/nederland-regionaal/geografische-data/wijk-en-buurtkaart-2020.

[23] R. Cervero, E. Guerra, and S. Al. *Beyond Mobility: Planning Cities for People and Places*. Island Press, 2017.

[24] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly Detection: A Survey". In: *ACM Computing Surveys* 41.3 (2009). DOI: 10.1145/1541880.1541882.

[25] Chung Chen and Lon-Mu Liu. "Joint Estimation of Model Parameters and Outlier Effects in Time Series". In: *Journal of the American Statistical Association* 88.421 (1993). URL: https://www.jstor.org/stable/2290724.

[26] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: ACM, 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. URL: http://doi.acm.org/10.1145/2939672.2939785.

[27] S. Christie and D. Fone. "Equity of Access to Tertiary Hospitals in Wales: a Travel Time Analysis". In: *Journal of Public Health* 25.4 (2003), pp. 344–350.

[28] David E. Clark, James M. McGibany, and Adam Myers. "The Effects of 9/11 on the Airline Travel Industry". In: *The Impact of 9/11 on Business and Economics: The Business of Terror*. Ed. by Matthew J. Morgan. New York: Palgrave Macmillan US, 2009, pp. 75–86. ISBN: 978-0-230-10006-0.

[29] M.C.G. Da Silva. "Measurements of Comfort in Vehicles". In: *Measurement Science and Technology* 13.6 (2002), R41.

[30] Maria Deutscher. "IBM's CEO Says Big Data is Like Oil, Enterprises Need Help Extracting the Value". In: *Silicon Angle* (2013). URL: https://siliconangle.com/2013/03/11/ibms-ceo-says-big-data-is-like-oil-enterprises-need-help-extracting-the-value.

[31] T.A. Domencich and D. McFadden. "Urban Travel Demand - a Behavioral Analysis". In: (1975).

[32] Ensheng Dong, Hongru Du, and Lauren Gardner. "An Interactive Web-Based Dashboard to Track COVID-19 in Real Time". English. In: *The Lancet Infectious Diseases* 20.5 (May 2020). DOI: 10.1016/S1473-3099(20)30120-1.

[33] Chotirat Ann Ratanamahatana Eamonn Keogh. "Exact Indexing of Dynamic Time Warping". In: *Knowledge and Information Systems* (2004). DOI: 10.1007/s10115-004-0154-9.

[34] John B. Edwards and Guy H. Orcutt. "Should Aggregation Prior to Estimation be the Rule?" In: *The Review of Economics and Statistics* 51.4 (1969), pp. 409–420. ISSN: 00346535, 15309142. URL: http://www.jstor.org/stable/1926432.

[35] Tim Erven and Jairo Cugliari. "Game-Theoretically Optimal Reconciliation of Contemporaneous Hierarchical Time Series Forecasts". In: *Lecture Notes in Statistics* 217 (Dec. 2013). DOI: 10.1007/978-3-319-18732-7_15.

[36] Gary W Evans and Richard E Wener. "Rail Commuting Duration and Passenger Stress." In: *Health psychology* 25.3 (2006), p. 408.

[37] Gary W Evans et al. "Community Noise Exposure and Stress in Children". In: *The Journal of the Acoustical Society of America* 109.3 (2001), pp. 1023–1027.

[38] Gene Fliedner. "An Investigation of Aggregate Variable Time Series Forecast Strategies with Specific Subaggregate Time Series Statistical Correlation". In: *Computers Operations Research* 26.10 (1999), pp. 1133–1149. ISSN: 0305-0548. DOI: https://doi.org/10.1016/S0305-0548(99)00017-9.

[39] Gene Fliedner. "Hierarchical Forecasting: Issues and Use Guidelines". In: *Ind. Manag. Data Syst.* 101 (2001), pp. 5–12.

[40] FlightRadar. *FlightRadar24*. 2020. URL: https://www.flightradar24.com.

[41] Howard Frumkin. "Urban Sprawl and Public Health". In: *Public Health Reports* (2016).

[42] Alex Gammerman et al. "Conformal k-NN Anomaly Detector for Univariate Data Streams". In: *Proceedings of Machine Learning Research* 60 (2017), pp. 1–15.

[43] Pablo García-Sánchez et al. "Testing Diversity-Enhancing Migration Policies for Hybrid On-Line Evolution of Robot Controllers". In: *European Conference on the Applications of Evolutionary Computation*. Berlin Heidelberg: Springer, 2012, pp. 52–62.

[44] Toni Giorgino. "Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package". In: *Journal of Statistical Software* 31.7 (2009), pp. 1–24. DOI: 10.18637/jss.v031.i07.

[45] David C Glass and Jerome E Singer. "Urban Stress: Experiments on Noise and Social Stressors". In: *Academic Press* (1972).

[46] Lei Gong et al. "Deriving Personal Trip Data from GPS Data: A Literature Review on the Existing Methodologies". In: *Procedia - Social and Behavioral Sciences* 138 (2014), pp. 557 –565.

[47] Google. *Google Trends: Explore what the World is Searching*. 2020. URL: https://trends.google.com.

[48] Yehuda Grunfeld and Zvi Griliches. "Is Aggregation Necessarily Bad?" In: *The Review of Economics and Statistics* 42.1 (1960), pp. 1–13. ISSN: 00346535, 15309142. URL: http://www.jstor.org/stable/1926089.

[49] Evert Haasdijk and Jacqueline Heinerman. "Quantifying Selection Pressure". In: *Evolutionary Computation* 26 (2018), pp. 213–235.

[50] Janine Hacker et al. "Virtually in this Together – How Web-Conferencing Systems enabled a New Virtual Togetherness During the COVID-19 Srisis". In: *European Journal of Information Systems* 29.5 (2020), pp. 563–584.

[51] J.A. Hanley and B.J. McNeil. "The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve". In: *Radiology* 143.1 (1982), 29–36.

[52] J Heinerman et al. "Can Social Learning Increase Learning Speed, Performance or Both?" In: *Proceedings of the 14th European Conference on Artificial Life ECAL 2017* (2017).

[53] Jacqueline Heinerman, Massimiliano Rango, and A. E. Eiben. "Evolution, Individual Learning, and Social Learning in a Swarm of Real Robots". In: *Proceedings - 2015 IEEE Symposium Series on Computational Intelligence, SSCI 2015*. IEEE. 2016, pp. 1055–1062. ISBN: 9781479975600. DOI: 10.1109/SSCI.2015.152.

[54] Jacqueline Heinerman et al. "Benefits of Social Learning in Physical Robots". English. In: *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018*. Institute of Electrical and Electronics Engineers Inc., 2019, pp. 851–858. DOI: 10.1109/SSCI.2018.8628857.

[55] Jacqueline Heinerman et al. "On-line Evolution of Foraging Behaviour in a Population of Real Robots". In: *European Conference on the Applications of Evolutionary Computation*. Springer, 2016, pp. 198–212.

[56] *HERE Technologies, 2020. Retrieved from https://developer.here.com/.*

[57] D. B. Hess. "Access to Employment for Adults in Poverty in the Buffalo-Niagara Region". In: *Urban Studies* 42.7 (2005), pp. 1177–1200.

[58] David Hojman and Robert F. K. Wynn. *Superb Forecasting or Self-Fulfilling Prophecy? The Economist on Thailand before the Asian Crisis*. Working Papers 2002_03. University of Liverpool, Department of Economics, 2002.

[59] Kirstin Hubrich. "Forecasting Euro Area Inflation: Does Aggregating Forecasts by HICP Component Improve Forecast Accuracy?" In: *International Journal of Forecasting* 21.1 (2005), pp. 119–136. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2004.04.005.

[60] Robert-Jan Huijsman, Evert Haasdijk, and A E Eiben. "An On-line On-board Distributed Algorithm for Evolutionary Robotics". In: *Artificial Evolution, 10th International Conference Evolution Artificielle*. Ed. by J-K. Hao et al. LNCS 7401. Springer, 2011, pp. 73–84.

[61] Humanitarian Data Exchange. *HDX: Global ports*. 2020. URL: https://data.humdata.org/dataset/global-ports.

[62] Rob Hyndman et al. "Optimal Combination Forecasts for Hierarchical Time Series". In: *Computational Statistics Data Analysis* 55 (Sept. 2011), pp. 2579–2589. DOI: 10.1016/j.csda.2011.03.006.

[63] Rob J Hyndman. *CRAN Task View: Time Series Analysis*. https://cran.r-project.org/web/views/TimeSeries.html. [Online; accessed 19-April-2021]. 2021.

[64] Rob J Hyndman and Yeasmin Khandakar. "Automatic Time Series Forecasting: the Forecast Package for R". In: *Journal of Statistical Software* 26.3 (2008), pp. 1–22. URL: https://www.jstatsoft.org/article/view/v027i03.

[65] Rob J. Hyndman, Alan J. Lee, and Earo Wang. "Fast Computation of Reconciled Forecasts for Hierarchical and Grouped Time Series". In: *Comput. Stat. Data Anal.* 97.C (2016), 16–32. ISSN: 0167-9473. DOI: 10.1016/j.csda.2015.11.007. URL: https://doi.org/10.1016/j.csda.2015.11.007.

[66] IATA. *What Can We Learn From Past Pandemic Episodes?* 2020. URL: https://www.iata.org/en/iata-repository/publications/economic-reports/what-can-we-learn-from-past-pandemic-episodes/.

[67] Domo Inc. *Data Never Sleeps 8.0*. https://www.domo.com/learn/infographic/data-never-sleeps-8. Infographic, accessed on 19-06-2021. 2021.

[68] International Energy Agency. *Changes in Transport Behaviour During the Covid-19 Crisis*. 2020. URL: https://www.iea.org/articles/changes-in-transport-behaviour-during-the-covid-19-crisis.

[69] Ben P Jolley, James M Borg, and Alastair Channon. "Analysis of Social Learning Strategies When Discovering and Maintaining Behaviours Inaccessible to Incremental Genetic Evolution". In: *International Conference on Simulation of Adaptive Behavior*. Springer, 2016, pp. 293–304.

[70] *Kadaster*. 2020. URL: https://www.kadaster.nl/zakelijk/registraties/basisregistraties/bag.

[71] Regina Kaiser and Agustin Maravall. "Seasonal Outliers in Time Series". In: *Statistics and Econometrics*. 15th ser. (1999).

[72] M. Kawabata. "Job Access and Employment among Low-Skilled Autoless Workers in US Metropolitan Areas". In: *Environment and Planning A* 35.9 (2003), pp. 1651–1668.

[73] M. Kawabata and Q. Shen. "Commuting Inequality between Cars and Public Transit: The Case of the San Francisco Bay Area, 1990-2000". In: *Urban Studies* 44.9 (2007), pp. 1759–1780.

[74] Avraham N Kluger. "Commute Variability and Strain". In: *Journal of Organizational Behavior: The International Journal of Industrial, Occupational and Organizational Psychology and Behavior* 19.2 (1998), pp. 147–165.

[75] Hannah Kollwitz and Alexis Papathanassis. "Evaluating Cruise Demand Forecasting Practices: A Delphi Approach". In: *Cruise Sector Challenges: Making Progress in an Uncertain World*. Wiesbaden: Gabler Verlag, 2011, pp. 39–55. ISBN: 978-3-8349-6871-5. DOI: 10.1007/978-3-8349-6871-5_3.

[76] Denis Kwiatkowski et al. "Testing the null Hypothesis of Stationarity Against the Alternative of a Unit Root". In: *Journal of Econometrics* 54.1 (1992), pp. 159–178.

[77] Javier López de Lacalle. *stsm: Structural Time Series Models*. R package version 1.9. 2016. URL: https://CRAN.R-project.org/package=stsm.

[78] A. Lavin and S. Ahmad. "Evaluating Real-time Anomaly Detection Algorithms – the Numenta Anomaly Benchmark". In: *14th International Conference on Machine Learning and Applications (IEEE ICMLA'15)* (2015).

[79] T. L. Lei and R. L. Church. "Mapping Transit-Based Access: Integrating GIS, Routes and Schedules". In: *International Journal of Geographical Information Science* 24.2 (2010), pp. 283–304.

[80] D. M. Levinson. "Accessibility and the Journey to Work". In: *Journal of Transport Geography* 6.1 (1998), pp. 11–21.

[81] *Lijst Emissiefactoren, Totale Lijst, 2020. Retrieved from https://www.co2emissiefactoren.nl/lijst-emissiefactoren*.

[82] S. Liu and X. Zhu. "Accessibility Analyst: an Integrated GIS Tool for Accessibility Analysis in Urban Transportation Planning". In: *Environment and Planning B: Planning and Design* 31.1 (2004), pp. 105–124.

[83] Xi Liu et al. "Revealing Travel Patterns and City Structure with Taxi Trip Data". In: *Journal of Transport Geography* 43 (2015), pp. 78 –90.

[84] Greta M. Ljung. "On Outlier Detection in Time Series". In: *Journal of the Royal Statistical Society* 55 (1993), pp. 559–567.

[85] A. Lovett et al. "Car Travel Time and Accessibility by Bus to General Practitioner Services: a Study Using Patient Registers and GIS". In: *Social Science & Medicine* 55.1 (2002), pp. 97–111.

[86] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[87] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "Predicting/Hypothesizing the Findings of the M5 Competition". In: *International Journal of Forecasting* (Nov. 2021). DOI: 10.1016/j.ijforecast.2021.09.014.

[88] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M4 Competition: Results, Findings, Conclusion and Way Forward". In: *International Journal of Forecasting* 34.4 (2018), pp. 802–808. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2018.06.001.

[89] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M5 competition: Background, Organization, and Implementation". In: *International Journal of Forecasting* (2021). ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2021.07.007.

[90] D. Martin, H. Jordan, and P. Roderick. "Taking the Bus: Incorporating Public Transport Timetable Data into Health Care Accessibility Modelling". In: *Environment and Planning A* 40.10 (2008), pp. 2510–2525.

[91] D. Martin et al. "Increasing the Sophistication of Access Measurement in a Rural Healthcare Study". In: *Health & place* 8.1 (2002), pp. 3–13.

[92] NDW. *Nationale Databank Wegverkeersgegevens, Open Data Portaal*. http://opendata.ndw.nu/. [Online; accessed 19-04-2021]. 2021.

[93] P. Nouvellet et al. "Reduction in Mobility and COVID-19 Transmission". In: *Nature Communications* 12 (2021).

[94] Paul J. O'Dowd, Matthew Studley, and Alan F T Winfield. "The Distributed Co-Evolution of an On-Board Simulator and Controller for Swarm Robot Behaviours". In: *Evolutionary Intelligence* 7.2 (2014), pp. 95–106. ISSN: 18645917. DOI: 10.1007/s12065-014-0112-8.

[95] Mark E Payton, Matthew H Greenstone, and Nathaniel Schenker. "Overlapping Confidence Intervals or Standard Error Intervals: What do they Mean in Terms of Statistical Significance?" In: *Journal of Insect Science* 3.1 (2003), p. 34.

[96] M.J. Pazzani and D. Billsus. "Content-Based Recommendation Systems". In: *The adaptive web*. Springer-Verlag, 2007, pp. 325–341.

[97] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[98] Diamantis Petropoulos Petalas, Hein van Schie, and Paul Hendriks Vettehen. "Forecasted Economic Change and the Self-Fulfilling Prophecy in Economic Decision-Making". In: *Plos One* 12.3 (2017), pp. 1–18. DOI: 10.1371/journal.pone.0174.

[99] Marco A.F. Pimentel et al. "A Review of Novelty Detection". In: *Signal Processing* 99 (2014), pp. 215–249. DOI: 10.1016/j.sigpro.2013.12.026.

[100] Haider Raza, Girijesh Prasad, and Yuhua Li. "EWMA Model Based Shift-Detection Methods for Detecting Covariate Shifts in non-Stationary Environments". In: *Pattern Recognition* 48 (2015), pp. 659–669.

[101] F. Ricci et al. *Recommender Systems Handbook*. Springer, 2011.

[102] Rijksoverheid. *Persconferenties Corona in Eenvoudige Taal*. 2020. URL: https://www.rijksoverheid.nl/onderwerpen/coronavirus-covid-19/vraag-en-antwoord.

[103] R.E. van Ruitenbeek and J.S. Slik. *Covid-19 cases, in Data Repository 'On the Relation between Covid-19, Mobility, and the Stock Market'*. 2021. DOI: https://doi.org/10.6084/m9.figshare.16989532.v1.

[104] R.E. van Ruitenbeek and J.S. Slik. *Covid-19 measures, in Data Repository 'On the Relation between Covid-19, Mobility, and the Stock Market'*. 2021. DOI: https://doi.org/10.6084/m9.figshare.16989535.v1.

[105] R.E. van Ruitenbeek and J.S. Slik. *Mobility, in Data Repository 'On the Relation between Covid-19, Mobility, and the Stock Market'*. 2021. DOI: https://doi.org/10.6084/m9.figshare.16989529.v1.

[106] R.E. van Ruitenbeek and J.S. Slik. *Readme, in Data Repository 'On the Relation between Covid-19, Mobility, and the Stock Market'*. 2021. DOI: https://doi.org/10.6084/m9.figshare.16989586.v1.

[107] R.E. van Ruitenbeek and J.S. Slik. *Stock prices, in Data Repository 'On the Relation between Covid-19, Mobility, and the Stock Market'*. 2021. DOI: https://doi.org/10.6084/m9.figshare.16989538.v1.

[108] R.E. van Ruitenbeek and J.S. Slik. *Stocks to watch, in Data Repository 'On the Relation between Covid-19, Mobility, and the Stock Market'*. 2021. DOI: https://doi.org/10.6084/m9.figshare.16989526.v1.

[109] Robin van Ruitenbeek, Jesper Slik, and Sandjai Bhulai. "On the Relation between Covid-19, Mobility, and the Stock Market". In: *PLOS ONE* (2021).

[110] Jens Schade and Bernhard Schlag. "Acceptability of Urban Transport Pricing Strategies". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 6.1 (2003), pp. 45–61.

[111] Anne-Maria Schweizer et al. "Outdoor Cycling Activity Affected by COVID-19 Related Epidemic-Control-Decisions". In: *Plos One* 16.5 (2021), e0249268.

[112] Skipper Seabold and Josef Perktold. "Statsmodels: Econometric and Statistical Modeling with Python". In: *9th Python in Science Conference*. 2010.

[113] Artemios-Anargyros Semenoglou et al. "Investigating the Accuracy of Cross-Learning Time Series Forecasting Methods". In: *International Journal of Forecasting* 37 (Dec. 2020). DOI: 10.1016/j.ijforecast.2020.11.009.

[114] G. Shani, R.I. Brafman, and D. Heckerman. "An MDP-Based Recommender System". In: *Journal of Machine Learning Research* 6 (2005).

[115] N. Sharma and P. Patterson. "The Impact of Communication Effectiveness and Service Quality on Relationship Commitment in Consumer, Professional Services". In: *Journal of Services Marketing* 13 (2 1999).

[116] Q. Shen. "A Spatial Analysis of Job Openings and Access in a US Metropolitan Area". In: *Journal of the American Planning Association* 67.1 (2001), pp. 53–68.

[117] C. Silva and P. Pinho. "The Structural Accessibility Layer (SAL): Revealing how Urban Structure Constrains Travel Choice". In: *Environment and Planning A* 42.11 (2010), pp. 2735–2752.

[118] Fernando Silva, Luís Correia, and Anders Lyhne Christensen. "A Case Study on the Scalability of Online Evolution of Robotic Controllers". In: *Portuguese Conference on Artificial Intelligence*. Switzerland: Springer International Publishing, 2015, pp. 189–200.

[119] Fernando Silva, Luís Correia, and Anders Lyhne Christensen. "Evolutionary Online Behaviour Learning and Adaptation in Real Robots". In: *Royal Society Open Science* 4.7 (2017), p. 160938.

[120] Fernando Silva et al. "odNEAT: An Algorithm for Distributed Online, Onboard Evolution of Robot Behaviours". In: *Artificial Life* 13 (2012), pp. 251–258.

[121] Filippo Simini et al. "A Universal Model for Mobility and Migration Patterns". In: *Nature* 484.7392 (2012), pp. 96–100.

[122] Jerome E Singer, Ulf Lundberg, and Marianne Frankenhaeuser. *Stress on the Train: A Study of Urban Commuting*. Psychological Laboratories, University of Stockholm, 1974.

[123] Uthayasankar Sivarajah et al. "Critical Analysis of Big Data Challenges and Analytical Methods". In: *Journal of Business Research* 70 (2017), pp. 263–286. ISSN: 0148-2963. DOI: https://doi.org/10.1016/j.jbusres.2016.08.001. URL: https://www.sciencedirect.com/science/article/pii/S014829631630488X.

[124] Jesper Slik and Sandjai Bhulai. "Accessibility Analysis for Private Car and Public Transport: Comparable Measures for Data-Driven Policymaking". In: (2021). Submitted for publication to the European Journal of Operations Research in December 2021.

[125] Jesper Slik and Sandjai Bhulai. "Approximate Dynamic Programming for Optimal Direct Marketing". In: *International Journal On Advances in Internet Technology* 13.1 and 2 (2020), pp. 65–72. ISSN: 1942-2652.

[126] Jesper Slik and Sandjai Bhulai. "Data-driven Direct Marketing via Approximate Dynamic Programming". In: *Proceedings of the 8th International Conference on Data Analytics (IARIA)* (2019), pp. 63–68. ISSN: 2308-4464.

[127] Jesper Slik and Sandjai Bhulai. "Detection of Additive Outliers in Univariate Time Series". In: (2021). Submitted for publication to Information Sciences in December 2021.

[128] Jesper Slik and Sandjai Bhulai. "Overcoming the Self-Fulfilling Prophecy in Time Series Forecasting". In: (2021). Submitted for publication to the International Journal of Forecasting in December 2021.

[129] Jesper Slik and Sandjai Bhulai. "Predicting Travel Behavior by Analyzing Mobility Transactions". In: *Journal of Traffic and Transportation Management* 2.2 (2020), pp. 25–33. ISSN: 2371-5782.

[130] Jesper Slik and Sandjai Bhulai. "Transaction-Driven Mobility Analysis for Travel Mode Choices". In: *The 11th International Conference on Ambient Systems, Networks and Technologies* 170 (2020), pp. 169–176. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2020.03.022.

[131] Jesper Slik and Sandjai Bhulai. "Understanding Human Mobility for Data-Driven Policy Making". In: (2021). Submitted for publication to Travel Behaviour and Society in December 2021.

[132] Taylor G. Smith et al. *pmdarima: ARIMA Estimators for Python*. [Online; accessed 24-11-2021]. 2017–. URL: http://www.alkaline-ml.com/pmdarima.

[133] Kenneth O Stanley and Risto Miikkulainen. "Evolving Neural Networks Through Augmenting Topologies". In: *Evolutionary Computation* 10.2 (2002), pp. 99–127.

[134] Statista. *Revenue from Big Data and Business Analytics Worldwide from 2015 to 2022*. https://www.statista.com/statistics/551501/worldwide-big-data-business-analytics-revenue/. Accessed on 28-06-2021. 2021.

[135] Statista. *Volume of Data/Information Created, Captured, Copied, and Consumed Worldwide from 2010 to 2025*. https://www.statista.com/statistics/871513/worldwide-data-created. Accessed on 28-06-2021. 2021.

[136] Samuel A. Stouffer. "Intervening Opportunities: A Theory Relating Mobility and Distance". In: *American Sociological Review* 5.6 (1940), pp. 845–867.

[137] Toyotaro Suzumura et al. "The Impact of COVID-19 on Flight Networks". In: *2020 IEEE International Conference on Big Data (Big Data)*. 2020, pp. 2443–2452.

[138] Alexander Szalay and Jim Gray. "Science in an Axponential World". In: *Nature* 440 (2006), pp. 413–414. DOI: 10.1038/440413a.

[139] Wesley Tansey, Eliana Feasley, and Risto Miikkulainen. "Accelerating Evolution via Egalitarian Social Learning". In: *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*. Ed. by Terence Soule. New York, NY, USA: ACM, 2012, pp. 919–926.

[140] Shannon Tellis. "Data is the 21st Century's Oil, says Siemens CEO Joe Kaeser". In: *The Economic Times India* (2018). URL: https://economictimes.indiatimes.com/magazines/panache/data-is-the-21st-centurys-oil-says-siemens-ceo-joe-kaeser/articleshow/64298125.cms.

[141] Alejandro Tirachini and Oded Cats. "COVID-19 and Public Transportation: Current Assessment, Prospects, and Research Needs". In: *Journal of Public Transportation* 22.1 (2020).

[142] Leo Tornqvist, Pentti Vartia, and Yrjo O. Vartia. "How Should Relative Changes Be Measured?" In: *The American Statistician* 39.1 (1985), pp. 43–46. URL: https://www.jstor.org/stable/2683905.

[143] *TravelTime Platform*. 2020. URL: https://www.traveltimeplatform.com/.

[144] "Universal predictability of mobility patterns in cities". In: *Journal of the Royal Society Interface* 11 (2014).

[145] Kuo-Ying Wang. "How Change of Public Transportation usage Reveals fear of the SARS Virus in a City". In: *Plos One* 9.3 (2014). DOI: 10.1371/journal.pone.0089405.

[146] Richard Wener and Gary W Evans. "Transportation and Health: the Impact of Commuting". In: *Encyclopedia of Environmental Health*. Elsevier Inc., 2011, pp. 400–407.

[147] Richard E Wener et al. "Running for the 7: 45: The Effects of Public Transit Improvements on Commuter Stress". In: *Transportation* 30.2 (2003), pp. 203–220.

[148] Darrell Whitley, Soraya Rana, and Robert B Heckendorn. "The Island Model Genetic Algorithm: On Separability, Population Size and Convergence". In: *Journal of Computing and Information Technology* 7.1 (1999), pp. 33–47.

[149] Shanika L. Wickramasuriya, George Athanasopoulos, and Rob J. Hyndman. "Optimal Forecast Reconciliation for Hierarchical and Grouped Time Series Through Trace Minimization". In: *Journal of the American Statistical Association* 114.526 (2019), pp. 804–819. DOI: 10.1080/01621459.2018.1448825.

[150] Wikipedia. *Busiest Airports by Continent*. 2020. URL: https://en.wikipedia.org/wiki/Busiest_airports_by_continent.

[151] G Williams, Jamie Murphy, and Rowena Hill. "A Latent Class Analysis of Commuters' Transportation Mode and Relationships with Commuter Stress". In: *Fourth International Conference on Traffic and Transport Psychology, Washington, DC*. 2008.

[152] K. O'Brien W.R. Richards and D.C. Miller. *New Air Traffic Surveillance Technology*. 2010. URL: http://www.boeing.com/commercial/aeromagazine/articles/qtr_02_10/pdfs/AERO_Q2-10_article02.pdf.

[153] YahooFinance. *Major World Indices*. 2020. URL: https://finance.yahoo.com/world-indices/.

[154] N. Yiannakoulias, W. Bland, and L. W. Svenson. "Estimating the Effect of Turn Penalties and Traffic Congestion on Measuring Spatial Accessibility to Primary Health Care". In: *Applied Geography* 39 (2013), pp. 172–182.

[155] George Kingsley Zipf. "The P1 P2/D Hypothesis: On the Intercity Movement of Persons". In: *American Sociological Review* 11.6 (1946), pp. 677–686.

Summary

Each minute in 2020, over 250,000 online meetings were held, more than 500 hours of video were uploaded, and USD 1M was spent online [67]. At the end of the year, over 64ZB of data were created [135]. Data is becoming a major part of our lives, and a continued growth is expected. In this dissertation, we demonstrate how to drive decisions based on data. We propose various methodology and contribute to different fields. Additionally, we implement part of the work through industry partnerships and achieve real-life results.

Data is the oil of the 21st century, according to various experts around the world [9, 140, 30]. Its promise is to support any organization in making better decisions. The comparison with a natural resource, like oil, seems to make sense, as data does little on its own. It needs to be converted to information, knowledge, or wisdom in order to deliver value. By 2022, organizations will have figured a way to deliver part of its promised value, resulting in an estimated 274 billion USD global industry [134].

However, unlike a natural resource, data is practically infinite, reusable, and becoming increasingly available. Therefore, novel challenges arise. A main, current challenge lies in identifying decisions and designing methodology for direct support. Often, substantial investment is required before the value of these supportive analyses is certain or even recognized. Auxiliary challenges include data integration, analytical skills, security and privacy, infrastructure, and synchronization [123].

Through nine independent chapters we propose methodology in which these challenges are tackled. The first three chapters of this dissertation concern *descriptive analyses*. These analyses aim to describe what happened. Intuitively, this seems a simple task, however, various technical or human errors can establish a challenge. The consecutive three chapters concern *predictive analyses*. These analyses build on descriptive analyses and aim to predict what is going to happen. A main challenge is to find meaningful, robust patterns and prevent overfitting. The final two chapters concern *prescriptive analyses*. These build on predictive analyses and aim to prescribe what to do. A main challenge is to balance the exploration of new knowledge and the exploitation of current knowledge in order to prescribe an optimal strategy.

After the introduction in Chapter 1, we develop novel methodology for detecting additive outliers in Chapter 2. We perceive an additive outlier as a surprisingly large or small value occurring for a single observation in a time series. The detection of these outliers is an important issue because their presence may have serious negative effects on the analysis in many different ways. Existing methods to detect such outliers are inadequate due to poor accuracy, high complexity, and long runtimes. In this research, we provide a novel approach to detect additive outliers that overcomes the mentioned drawbacks. We validate our approach by comparing against

existing techniques and benchmark performance. Experimental results on benchmark datasets show that our proposed technique outperforms existing methods on several measures.

In Chapter 3, we aim to identify the patterns of behavior which underlie human mobility. More specifically, we compare commuters who drive in a car with those who use the train in the same geographic region of the Netherlands. We try to understand the mode choices of the commuters based on three factors: the cost of the transport mode, the CO_2 emissions, and the travel time. The analysis has been based on data consisting of travel transactions in the Netherlands during 2018 containing over half a million records. The results can be used to stimulate behavioral change proactively. Moreover, the data and results can also be utilized to improve trip planners.

In Chapter 4, we argue the Covid-19 pandemic has brought forth a major landscape shock in the mobility sector. Due to its recentness, researchers have just started studying and understanding the implications of this crisis on mobility. We contribute by combining mobility data from various sources to bring a novel angle to understanding mobility patterns during Covid-19. The goal is to expose relations between the variables and understand them by using our data. This is crucial information for governments to understand and address the underlying root causes of the impact.

In Chapter 5, we argue that urban planning can benefit tremendously from a better understanding of where, when, why, and how people travel. Through advances in technology, detailed data on the travel behavior of individuals has become available. This data can be leveraged to understand why one prefers one mode of transportation over another. We analyze a unique dataset through which we can address this question. We show that the travel behavior in our dataset is highly predictable, with an accuracy of 97%. The main predictors are reachability features, more so than specific travel times. Moreover, the travel type (commute or personal) has a considerable influence on travel mode choice.

In Chapter 6, we argue that the disparity between the accessibility of areas through different travel modes is essential for the choice of the mode of transport. Calculation of the travel times by different travel modes is, therefore, very important. Many urban design decisions on infrastructure depend on these calculations. Developments in open data policies among urban data producers make this analysis more tractable. In this chapter, we apply a data-driven approach to travel time estimation based on realized past travel times. We compare commuters who drive in a car with those who use the train in the same geographic region of the Netherlands. First, we propose a method to quantify the accessibility of areas for these different modalities. Second, we show how these metrics can be used to determine optimal locations based on the willingness to travel. The results can be integrated into planning software to making data-driving decisions for policymaking.

In Chapter 7, we observe two current challenges in time series forecasting are the self-fulfilling prophecy and finding robust seasonal patterns. We argue that both can be overcome through combining similar time series. We propose methodology to extract robust seasonal patterns from low-level sales data through applying hierarchical clustering. We validate our approach using a simulation experiment and a real-life dataset containing over €2B of bicycle sales. Our simulation results show a 45% decrease in forecasting error and they quantify the effects of the self-fulfilling prophecy on forecasting error. Our results on real-life data show a 15% performance

gain on the benchmark when applying clustering. Additionally, we show insights on the effects of applying smoothing and forecasting sell-in vs sell-out data.

In Chapter 8, we argue that email marketing is a widely used business tool that is in danger of being overrun by unwanted commercial email. Therefore, direct marketing via email is usually seen as notoriously difficult. One needs to decide which email to send at what time to which customer in order to maximize the email interaction rate. Two main perspectives can be distinguished: scoring the relevancy of each email and sending the most relevant, or seeing the problem as a sequential decision problem and sending emails according to a multi-stage strategy. In this chapter, we adopt the second approach and model the problem as a Markov decision problem (MDP). The advantage of this approach is that it can balance short- and long-term rewards and allows for complex strategies. We illustrate how the problem can be modeled such that the MDP remains tractable for large datasets. Furthermore, we numerically demonstrate by using real data that the optimal strategy has a high interaction probability, which is much higher than a greedy strategy or a random strategy. Therefore, the model leads to better relevancy to the customer and thereby generates more revenue for the company.

In Chapter 9, we focus on robot-to-robot learning. This is a specific case of social learning in robotics that enables the ability to transfer robot controllers directly from one robot to another. Previous studies showed that the exchange of controller information can increase learning speed and performance. However, most of these studies have been performed in simulation, where robots are identical. Therefore, the results do not necessarily transfer to a real environment, where each robot is unique per definition due to the random differences in hardware. In this research, we investigate the effect of exchanging controller information, on top of individual learning, in a group of Thymio II robots for two tasks: obstacle avoidance and foraging. The controllers of the robots are neural networks that evolve using a modified version of the state-of-the-art NEAT algorithm, called *cNEAT*, which allows the conversion of innovations numbers from other robots. This research shows that robot-to-robot learning seems to at least parallelize the search, reducing wall clock time. Additionally, controllers are less complex, resulting in a smaller search space.

Samenvatting

Tijdens iedere minuut van 2020 zijn er meer dan 250.000 online vergaderingen gehouden, is er meer dan 500 uur aan video geüpload, en werd er USD 1M aan online aankopen gedaan [67]. Aan het einde van het jaar is er meer dan 64ZB aan data gecreëerd [135]. Data is een groot gedeelte van ons leven aan het worden, en de verwachting is dat dit alleen maar meer zal worden in de komende jaren. In deze thesis laten wij zijn hoe beslissingen op data gebaseerd kunnen worden. We stellen verscheidene methodologie voor en dragen hiermee aan bij binnen verschillende onderzoeksvelden. Hiernaast implementeren wij dit werk door middel van samenwerkingen met de industrie, en leveren resultaten in het echte leven.

Volgens meerdere experts in de wereld is data de olie van de 21^{e} eeuw [9, 140, 30]. De belofte is dat data iedere organisatie kan ondersteunen in de besluitvorming. De vergelijking met een natuurlijke grondstof als olie klinkt logisch, omdat data uit zichzelf niets doet. Het moet worden geconverteerd naar informatie, kennis en wijsheid om er waarde uit te halen. In 2022 is verwacht dat organisaties een gedeelte van deze waarde reeds geconverteerd heeft, wat zal resulteren in een mondiale industrie van USD 274 miljard [134].

Echter, in tegenstelling tot een natuurlijke grondstof, is data praktisch gezien oneindig, herbruikbaar, en komt er meer en meer beschikbaar. Hierdoor ontstaan nieuwe uitdagingen. Een grote, hedendaagse uitdaging is om beslissingen te identificeren en methodologie te ontwerpen welke direct met data ondersteund wordt. Hiernaast is er vaak een grote investering nodig voordat de waarde van deze analyses bewezen of erkend is en zijn er uitdagingen in integratie, vaardigheden, veiligheid en privacy, infrastructuur, en synchronisatie [123].

Door negen onafhankelijke hoofdstukken stellen wij methodologie voor waarin deze uitdagingen overkomen worden. De eerste drie hoofdstukken van deze thesis betreffen *beschrijvende analyses*. Deze analyses proberen zo goed mogelijk te beschrijven wat er gebeurd. Dit lijkt een makkelijke taak, echter, verscheidene technische of menselijke fouten kunnen tot problemen leiden. De volgende drie hoofdstukken betreffen *voorspellende analyses*. Deze analyses bouwen verder op beschrijvende analyses, en proberen vervolgens te voorspellen wat er zal gaan gebeuren. Een uitdaging hierin is het vinden van veelzeggende en robuuste patronen in de data. De laatste twee hoofdstukken betreffen *voorschrijvende analyses*. Deze analyses bouwen verder op voorspellende analyses, en proberen te voorschrijven wat het beste gedaan kan worden. Een grote uitdaging hierin is het balanceren van opgedane kennis en het vergaren van nieuwe kennis, om zo tot een optimale strategie te komen.

Na de introductie in hoofdstuk 1, ontwikkelen wij een nieuwe methodologie om additieve uitschieters te detecteren in hoofdstuk 2. Wij interpreteren een additieve uitschieter als een verrassend grote of kleine waarde welke gerepresenteerd wordt door een enkele observatie in een tijdreeks. Het detecteren hiervan is belangrijk, gezien ze serieuze consequenties kunnen hebben op de analyse van deze

data. Huidige methoden om deze te detecteren zijn niet toereikend, doordat de nauwkeurigheid niet goed genoeg is, de complexiteit hoog is, en ze een lange rekentijd vergen. In dit onderzoek presenteren wij een nieuwe aanpak om de additieve uitschieters te detecteren, waarin de genoemde tekortkomingen worden overwonnen. Onze aanpak valideren wij door een vergelijking te maken met huidige methodologie op verschillende benchmarks. De experimentele resultaten laten zien dat onze methode de huidige methodologie op verschillende vlakken verslaat.

In hoofdstuk 3 identificeren wij de gedragspatronen die onderliggend zijn aan menselijke mobiliteit. We vergelijken woon-werkverkeer tussen auto- en treingebruikers in hetzelfde geografische gebied in Nederland. Wij proberen de keuze voor het type mobiliteit te begrijpen op basis van drie factoren: de reiskosten, de CO_2 uitstoot, en de reistijd. De analyse is gebaseerd op reis transacties in Nederland in 2018, en bestaan uit een half miljoen gemaakte reizen. De resultaten kunnen worden gebruikt om gedragsverandering proactief te stimuleren. Hiernaast kunnen de resultaten worden gebruikt om routeplanners te verbeteren.

In hoofdstuk 4 laten wij zien dat de Covid-19 pandemie een grote schok in het mobiliteitslandschap teweeg heeft gebracht. Gezien de recentheid zijn onderzoekers net begonnen met het bestuderen en begrijpen van de implicaties van de crisis op mobiliteit. Wij dragen hieraan bij door mobiliteitsdata op verschillende vlakken van verschillende bronnen bijeen te brengen, en een nieuw perspectief te geven op de patronen tijdens de pandemie. Het doel is om relaties tussen verschillende variabelen naar boven te brengen en deze te begrijpen op basis van onze data. Deze relaties zijn cruciaal voor overheden om de gevolgen van de pandemie te begrijpen.

In hoofdstuk 5 beargumenteren wij dat stedelijke planning enorm kan profiteren door beter te begrijpen waar, wanneer, waarom, en hoe mensen reizen. Door ontwikkelingen in technologie is gedetailleerde data over het mobiliteitsgedrag van gebruikers beschikbaar geworden. Deze data kan worden gebruikt om beter te begrijpen waarom de ene modaliteit wordt geprefereerd over de andere. Wij analyseren een unieke dataset waarmee wij deze vraag kunnen beantwoorden. Wij laten zien dat het reisgedrag in onze data erg voorspelbaar is, met een nauwkeurigheid van 97%. De belangrijkste factoren hiervoor zijn de bereikbaarheid van verschillende modaliteiten. Deze worden geprefereerd boven specifieke reistijden. Hiernaast heeft het reistype (persoonlijk of woon-werk) een grote invloed op de modaliteitskeuze.

In hoofdstuk 6 betuigen wij dat het verschil tussen de toegankelijkheid tussen verschillende gebieden op het gebied van modaliteit essentieel is voor de modaliteitskeuze. Berekeningen van reistijden voor verschillende modaliteiten is daarom erg belangrijk. Vele keuzes met betrekking tot stedelijke ontwikkeling en infrastructuur zijn afhankelijk van deze berekeningen. Ontwikkelingen in open data beleid tussen stedelijke data producenten maken deze analyse handelbaar. In dit hoofdstuk passen wij een data gedreven aanpak toe op reistijd inschattingen, gebaseerd op gerealiseerde reistijden. Wij vergelijken woon-werkverkeer tussen de modaliteiten auto en trein in dezelfde regio in Nederland. Ten eerste stellen wij een methode voor om de bereikbaarheid van gebieden met verschillende modaliteiten te kwantificeren. Ten tweede laten wij zien hoe deze statistieken kunnen worden gebruikt om optimale locaties te bepalen voor nieuwe typen vervoer gebaseerd op de bereidbaarheid om te reizen. De resultaten kunnen worden geïntegreerd in planning software en om data-gedreven beleid op te kunnen stellen.

In hoofdstuk 7 observeren wij twee uitdagingen in hedendaagse forecasting algoritmen: het vinden van robuuste seizoenspatronen en het overkomen van de zelf-vervullende voorspelling. Wij beargumenteren dat beiden overkomen kunnen worden door vergelijkbare tijdreeksen met elkaar te combineren. Wij introduceren methodologie om robuuste seizoenspatronen uit gedetailleerdere tijdreeksen te extraheren door hiërarchisch clusteren toe te passen. Onze aanpak valideren wij door een simulatie experiment en op een dataset gegenereerd in het echte leven die meer dan €2 miljard aan fiets verkopen bevat. Onze resultaten uit de simulatie demonstreren een afname van 45% in de fout van de voorspellingen en ze kwantificeren het effect van de zelf-vervullende voorspellingen. De resultaten op de empirische dataset laten een 15% verbetering zien ten opzichte van de benchmark, wanneer clustering toegepast wordt. Hiernaast demonstreren wij inzichten in het toepassen van smoothing en het verschil tussen voorspellingen op sell-in en sell-out.

In hoofdstuk 8 beargumenteren wij dat e-mail marketing een veelgebruikte tool is en dat hierin het gevaar van ongewenste commerciële email schuilt. Hierdoor wordt directe email marketing gezien als een moeilijke taak. Er moet besloten worden welke email op welk tijdstip naar welke persoon gestuurd moet worden, om zo tot een effectieve marketingstrategie te komen. Hier bestaan twee strategieën voor: de relevantie van iedere email voor iedere persoon scoren en de meest relevante sturen, of het probleem zien als een sequentieel beslisprobleem en emails sturen volgens een lange termijn strategie. In dit hoofdstuk adopteren wij de tweede aanpak en modelleren wij het probleem als een Markov Decision Problem (MDP). Het voordeel van deze aanpak is dat het korte- en lange termijn feedback kan balanceren en dat het geavanceerde strategieën kan vinden. Wij illustreren hoe het probleem op een manier kan worden geformuleerd dat het handelbaar is voor grote datasets. Hiernaast demonstreren wij op echte data dat de optimale strategie een hoge interactie kans geeft, eentje veel hoger dan een korte termijn of willekeurige strategie. Doordat het model relevantere emails voorschotelt aan de consumenten, geeft het ieder bedrijf een kans op hogere relevantie tot de klant en hierdoor de kans op meer inkomsten.

In hoofdstuk 9 focussen wij op 'robot-to-robot learning'. Dit is een specifiek geval van sociaal leren binnen robotica, welke de robot de mogelijkheid geeft om controllers direct uit te wisselen tussen elkaar. Voorgaande studies hebben gedemonstreerd dat deze uitwisseling van controllers het leerproces en de prestaties kan versnellen. Echter zijn de meeste van deze studies uitgevoerd in een simulatie, waar iedere robot identiek is. Hierdoor gelden de resultaten niet direct naar een echte omgeving, waar iedere robot uniek is per definitie door verschillen in de hardware. In dit hoofdstuk onderzoeken wij het effect van het uitwisselen van de controllers, bovenop 'individual learning', in een groep van Thymio II robots voor twee taken: obstakel vermijding en foerageren. De controllers van de robots zijn neurale netwerken welke evolueren door middel van een aangepaste versie van het NEAT algoritme, genaamd *cNEAT*, welke de innovatiegetallen tussen de robots kan converteren. Het onderzoek laat zien dat het leren tussen robots tenminste geparallelliseerd wordt, wat de doorlooptijd reduceert. Hiernaast zijn de controllers minder complex, wat resulteert in een kleinere zoekruimte.