Enhancing bicycle route choice predictions:
A calibration tool for the Fietsmonitor

BSc thesis Civil Engineering
Department of Engineering Technology

Author: Judith van der Velden
Student number: S2843609
Supervisor Witteveen+Bos: Sander Veenstra
First supervisor UT: Anika Laschewski
Second supervisor UT: Monica Pena Acosta

June 18, 2025

(image by: Mobiliteit.nl (2024))

Preface

This thesis represents the final step of my bachelor in Civil Engineering here at the University of Twente. I carried out this thesis in collaboration with Witteveen+Bos, in the 'Verkeer en Mobiliteitsdata' department, to be exact. Over the past ten weeks, I have spent many hours learning how to carry out such a large research project on my own. Conducting this research project has taught me valuable skills such as time management and computer programming that I will take with me for the rest of my career.

I want to thank Witteveen+Bos for the opportunity to conduct this research project at their company. In particular, I want to thank my external supervisor, Sander Veenstra, who has guided me at the company over the past weeks. Not only has he shared much of his expertise regarding the Fietsmonitor itself and the Python script that came with it, but he has also provided me with valuable insights regarding the development of the calibration tool and feedback on my report. On top of that, he made sure I felt at home in the office and got to know many of the other employees in the department.

Furthermore, I would like to thank my internal supervisor, Anika Laschewski, for all the fast responses to my questions regarding my thesis and all the feedback sessions throughout my graduation period. These sessions have guided my thesis towards the final product that you are about to read.

I hope you enjoy reading my thesis as much as I enjoyed writing it.
Sincerely,

Judith van der Velden
Enschede, June 2025

Executive summary

Around 28% of the total trips made by Dutch citizens are made by bicycle (de Haas and Kolkowski, (2023)). Because of this, bicycle traffic must be taken into consideration during the decision process regarding (bicycle) infrastructure. Witteveen+Bos has developed the Fietsmonitor to assist stakeholders in this process.

The Fietsmonitor is a computer tool that makes predictions on the bicycle traffic in a city using OD-matrices and a 'shortest-path' route choice model. Previous studies, such as Van Nijen et al., (2024), have found that many more decision variables also impact route choice besides travel distance. This knowledge called for an improvement of the Fietsmonitor: a calibration tool that connects the traffic predictions to the observed bicycle traffic in a city.

In this thesis, a calibration tool for the Fietsmonitor is developed, while taking into account the uncertainties in the available input data: the OD-matrix and the observed count data. The calibration tool uses a machine learning approach through gradient descent in order to optimize the uncertain parameters. It does this through four different types of calibration: the route choice probabilities, the traffic within each OD-pair, the overall OD-matrix, and the observed count data.

The model performs best if the route choice probabilities are initialized as equal over the different routes, the probabilities are normalized through a standard normalization function, and a learning rate of 0.1 is used. A sensitivity analysis has shown that the calibration tool is robust to changes in the input data.

In the end, the calibration tool managed to bring a mean absolute error of 610 bicycles down to a mean absolute error of only 64 bicycles in 100 iterations. This is a significantly better prediction of the bicycle traffic compared to the original Fietsmonitor. Thus, the developed calibration tool was found to be a valuable addition to the Fietsmonitor.

Contents

1 Introduction

For years now, the Netherlands has had a reputation as a bicycle country, and not for no good reason: the Dutch make over a quarter (28%) of their trips by bicycle (de Haas and Kolkowski, (2023)). Due to the high volume of bicycle traffic on roads, infrastructural decisions cannot be made without considering bicycle traffic.

To help municipalities and other large decision makers in this process, Witteveen+Bos has developed the Fietsmonitor. The Fietsmonitor gives stakeholders insight into the bicycle traffic load on the network, which allows them to make substantiated decisions regarding bicycle infrastructure (Veenstra, (*n.d.*)).

One of the main weaknesses of the current version of the Fietsmonitor is its traffic assignment method. Currently, it uses a "shortest-path" route choice model, even though it has been proven in the past that cyclists do not always choose the shortest route to travel from origin to destination. A study by Delft University of Technology, for example, has found that as little as 32.6% of cyclists choose the shortest route over longer alternatives (Ton et al., (2017)).

Multiple studies have found that many more decision variables affect route choice, rather than just travel distance. Van Nijen et al., (2024) has identified 16 different spatial decision variables and their impact on the route choice decision process. Next to these spatial characteristics, Harms et al., (2014) notes that social characteristics and opportunities also impact the decision-making process.

From these decision variables can be concluded that the current "shortest-path" route choice method is an insufficient representation of the real-life traffic situation. This is supported by the gaps between the predicted traffic by the Fietsmonitor and the observed traffic in the study area. In this thesis, a calibration tool will be developed in order to improve the route assignment method of the Fietsmonitor. This new calibration tool aims to provide stakeholders with a more accurate reflection of the bicycle traffic situation in a city, and can help them make more suitable infrastructural design choices that fit the needs and behavior of cyclists.

Next to the Fietsmonitor, other studies have attempted to predict bicycle route choice by various means. These studies often try to derive behavioral characteristics from the input data. For example, Meister et al., (2023) and Łukawska et al., (2023) both try to predict bicycle route choice based on GPS trajectory input data. These studies differ from the research project at hand in two ways: the Fietsmonitor does not use GPS trajectory data in its modeling process, and the goal of the calibration in this thesis is to close the gap between predicted and observed bicycle traffic, rather than predicting route choice behavior from the model.

1.1 The Fietsmonitor

The Fietsmonitor is a computer tool developed by Witteveen+Bos that predicts bicycle traffic in a city. Over the past years, the Fietsmonitor has been used in multiple cities throughout the Netherlands with a number of different objectives. Its main purpose is to help stakeholders make substantiated design choices in the bicycle infrastructural network. In Haarlem, for example, the Fietsmonitor was used to identify bicycle intensities, characteristics of existing bicycle infrastructure, and possible problem areas regarding bicycle safety and road design (Witteveen+Bos, (*n.d.*)).

The Fietsmonitor's main input data are OD-matrices (origin-destination-matrices), which are either open source or provided by the stakeholders. The original version of the Fietsmonitor uses a shortest path route choice model: it calculates the shortest path from each origin to each destination, and models all traffic from the OD-pair over this route.

Recently, an additional function for the Fietsmonitor has been developed. This function constructs a set number of routes from each origin to each destination and divides the traffic from the OD-matrix over these routes. The route set is constructed through the Dijkstra algorithm. In the current version, three routes are constructed using 0.5, 0.3, and 0.2 as the increments. After the shortest route for an OD-pair has been constructed, the road segments in the route are given a penalty, 0.5 in this case. This makes the route less feasible to use during the construction of the second shortest route. The second shortest route is then given a penalty of 0.3, and lastly, the third shortest route is made.

The traffic over the routes is divided through the increments that are used to construct the routes. This division can be seen as the probability of choosing a route.

All bicycle traffic from all origins to all destinations is then projected onto the road network of the study area and summed to gain insight into the overall traffic intensity on the road segments (named 'links' in the Fietsmonitor). This can then be used by stakeholders to gain improved insight into the bicycle traffic intensity in the city of interest.

1.2 Thesis outline

In the following chapters of this report, the research project is conducted step-by-step. First, section 2 discusses the problem context to gain insight into the current state of the Fietsmonitor, and how the goal of this research project, the development of the calibration tool, can be achieved. Section 3 takes the first steps towards the development of the calibration tool by diving into the theoretical background on model calibration and machine learning. Using this knowledge, the methodology chapter, section 4, explains how the model is to be developed, as well as how the data validation and the sensitivity analysis are carried out. In section 5, the results of the model and the sensitivity analysis are presented. This outcome is discussed in section 6. Conclusions are drawn and the research questions are answered in section 7. The final chapter of this report, section 8, gives recommendations for possible future research that might build upon the research conducted in this thesis.

2 Problem context

In this chapter of the report, some context is given regarding this research project. To start off, this chapter discusses the problem of the current version of the Fietsmonitor, and issues in the available input data. After this, the research aim and research questions are introduced. Lastly, the project is demarcated through the project scope and study area.

2.1 Problem statement

As introduced in section 1.1, the Fietsmonitor uses OD-matrices and a route choice probability division to model traffic over the bicycle infrastructure of a city. The model does not account for any route choice decision variables besides travel distance. As stated in section 1, this is insufficient for a realistic traffic division. This causes significant gaps between the predictions made by the Fietsmonitor and the observed traffic.

The OD-matrix used to create the traffic in this thesis is sourced from ODiN (OnDerweg in Nederland), which provides traffic data on the daily travel behavior of citizens in the Netherlands (Centraal Bureau voor de Statistiek, (2024)). OD-matrices are available through ODiN, but these do not represent the entire population of a city. The CBS (Centraal Bureau voor de Statistiek) collects data through a sample survey over 30 different citizen categories (based on age, origin, and income) to get an estimate of the average travel behavior of Dutch citizens.

In order to account for this small sample size, the Fietsmonitor combines the traffic from these OD-matrices with the average amount of trips by bicycle per person per day, in order to estimate the OD-matrix if all citizens were included. Note that the traffic per person per day is an average, and may not reflect the true amount of bicycle traffic generated in a zone of the study area on any given day.

In order to determine how correct the predictions made by the model are compared to the real-life bicycle traffic, count data is used. For this, the model uses data sourced from NDW (Nationaal Dataportaal Wegverkeer). NDW is an open source of traffic data in the Netherlands provided by the Dutch government (Wegverkeer, (*n.d.*)).

This data is not entirely reliable either: there are large fluctuations in the amount of count data available between the different locations, the latest count data available in terms of time differs by over 2 years between locations, and some count locations have a standard deviation between days of over 1000 cyclists. Lastly, there may have been external factors, such as heavy rainfall or long-term road work, that may have affected the travel behavior of cyclists during the data collection period.

From the issues mentioned above can be concluded that the input data used in the Fietsmonitor is not entirely reliable. A solution for these issues needs to be integrated within the calibration tool, while bringing the predicted traffic in line with the observed traffic. This can be done by not only calibrating the probabilities of choosing the routes, but also making alterations in the OD-matrix and the count data.

2.2 Research objective and boundaries

2.2.1 Research aim

The aim of this thesis is to develop a calibration tool for the bicycle traffic predictions made by the Fietsmonitor. The calibration tool should only take an OD-matrix and count data as its input. This calibration method should calibrate the probability of choosing the shortest, second shortest or third shortest route from origin to destination in order to improve the bicycle traffic predictions in the study area. Here, it is also important to pay attention to the uncertainties within the available input data by calibrating the OD-matrix and count data alongside the route choice probabilities. The calibration tool should not only perform well in terms of outcome, but should also be able to converge within a reasonable processing time.

2.2.2 Research questions

In order to arrive at the research aim, a number of questions have been drawn up. The main research question to be answered by the content of this thesis is:

"How can bicycle route choice predictions in the Fietsmonitor be calibrated, using OD-matrices and count data as inputs?"

In order to answer this more complex question, it has been broken down into sub-questions. These questions discuss previously conducted research and apply the gained knowledge to the study area. The sub-questions for this research project are:

- **Q1:** What calibration methods exist and what method of calibration would be most appropriate in the case of the Fietsmonitor, given the reliability of the OD-matrix and count data?
- **Q2:** What assignment methods are possible, and how can bicycle streams best be assigned over different routes for the use of the calibration tool?

These first two research questions will mainly operate as guidelines during the development process of the calibration tool. They provide background knowledge and allow for substantiated decision-making in the development stage, after which the last two research questions can be answered.

- **Q3:** What does the calibrated route choice model from question 2 look like?
- **Q4:** How sensitive is the calibration tool to changes in the input data?

Using the knowledge gathered from sub-questions 1 and 2, the model itself can be developed. These last two questions dive deeper into the optimization of the model to get it to perform optimally.

2.2.3 Scope

The model will calibrate the bicycle traffic predictions made by the Fietsmonitor. For this, only the OD-matrices and count data as discussed in section 2.1 will be used. Decision variables such as the ones discussed in section 1 are outside the scope of this thesis.

Temporal factors such as the time of the day or the difference between bicycle traffic during weekdays and weekends are also not taken into account. The model works with an average amount of traffic on an average day of the week.

2.2.4 Study area

The study area for the development of the calibration tool is Apeldoorn. This city has a lot of count data available at many locations, which is essential for the calibration process. Some of these count locations lie in villages around Apeldoorn. However, the OD-matrices only provide information on trips made from and to Apeldoorn, whereas these villages may also very well generate their own traffic or receive traffic from other nearby villages or cities. Thus, making true predictions on the amount of traffic in these locations based on the OD-matrices is not possible. Therefore, it was decided to demarcate the city to the inner city and the near suburbs. The demarcation is presented in Figure 1.

This demarcation of the study area means that some of the available count locations are left out of the study area. Figure 2 shows the demarcated study area compared to all available count locations. Count locations on the border of the study area are considered to be part of the study area, all other locations outside the border are not considered as part of this study.

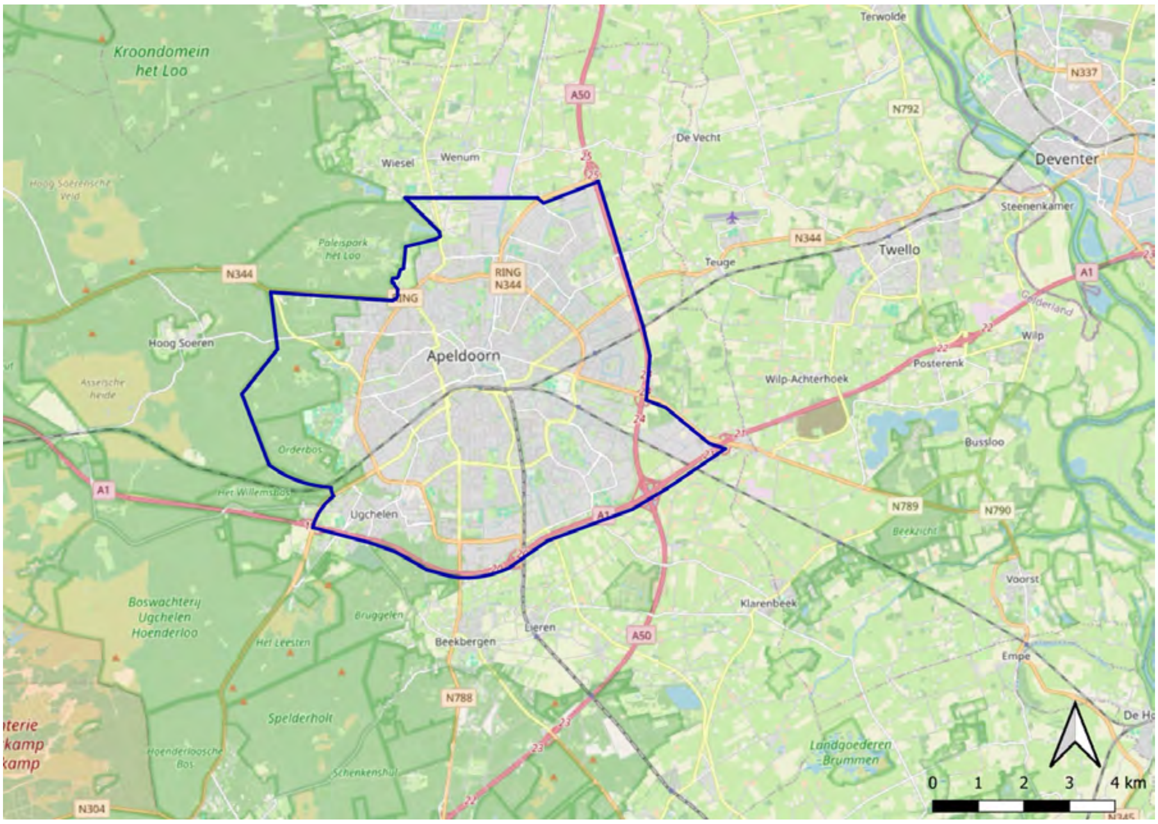

Figure 1: Study area of this research project: Apeldoorn (OpenStreetMap, (*n.d.*))

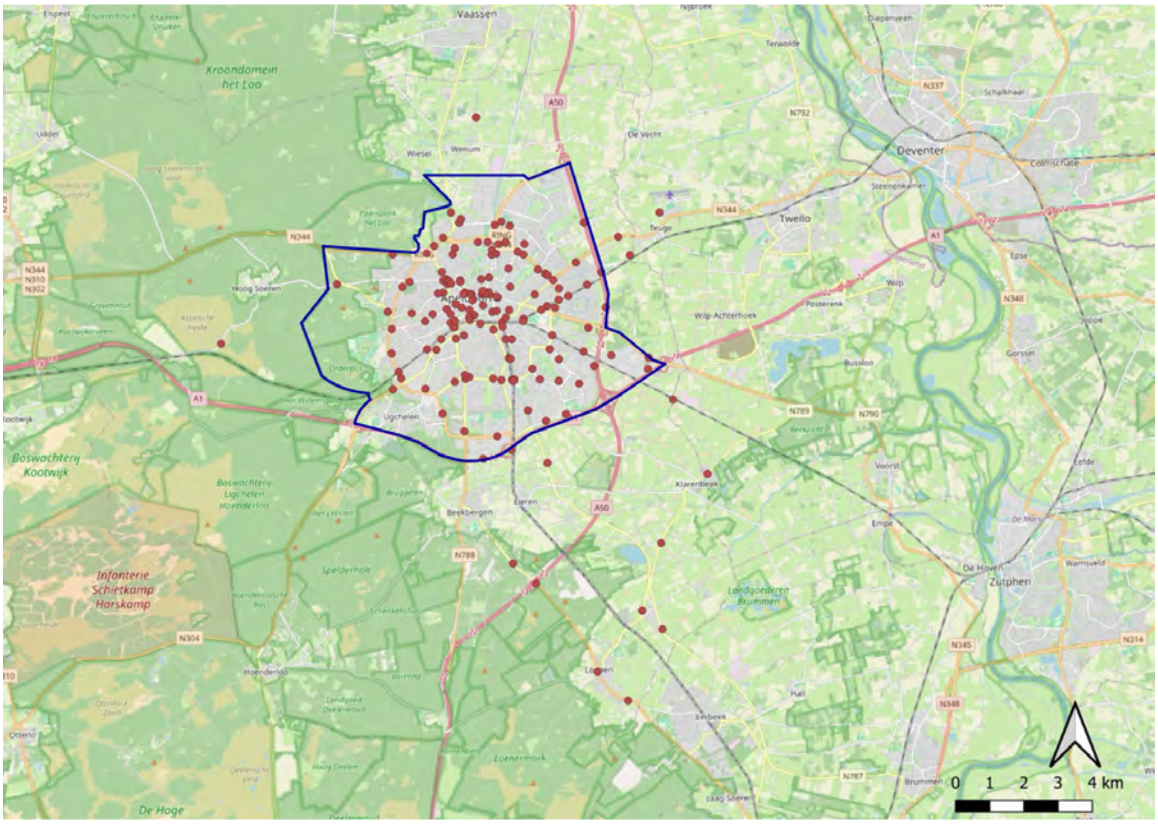

Figure 2: Count locations in the study area (OpenStreetMap, (*n.d.*))

3 Theoretical Background

In this chapter, relevant background knowledge is introduced and elaborated on. This knowledge is necessary in order to understand the methodology as discussed in section 4. First, model calibration itself is introduced by diving deeper into the different methods and measures for model calibration. After this, an introduction is made on machine learning, and the different relevant elements are elaborated on.

3.1 Model calibration

This section of the theoretical background dives deeper into the workings of model calibration. In order to go through this step-wise, first, the different kinds of classification processes are introduced, after which different measures of calibration are discussed.

3.1.1 Classification methods

Calibration methods are applied to classification processes, which are methods of sorting data into classes that have been determined beforehand (Keylabs, (2024)). The three most well-known methods, which will also be considered for this research project, are (Deepchecks, (2022)):

- **Binary classification** is used in the situation where there are only two classes a case can be put into. An example may be whether or not a person travels by bicycle.
- **Multi-class classification** refers to the situation where there are multiple classes a case can be put into. Here, an example may be that someone travels by bicycle, car, public transport, or by foot.
- **Multi-label classification**, where multiple classes may be relevant for each individual case. For example, it may now be possible that the individual walks to a certain location and takes public transport from there.

The calibration tool for the Fietsmonitor needs to allow the individual traveling by bicycle to choose between a number of different routes. The Fietsmonitor itself does not necessarily need to allow the cyclists to change from one route to the other halfway through. This is not taken into consideration during the construction of the route set from each origin to each destination, and in real life, only very few cyclist will switch between routes during their travel. Thus, the calibration tool for the Fietsmonitor fits the multi-class classification process.

3.1.2 Calibration measures

The Fietsmonitor tries to predict the number of cyclists on each link in the network. Calibration methods are used to measure and improve such predictions to get the predictions in line with the actual observed value of that variable (Deepchecks, (2024)). In the case of the calibration tool for the Fietsmonitor, this would mean that if the computer model predicts, for example, a 30% probability that a cyclist chooses the shortest route between origin and destination rather than one of two slightly longer routes, this would also be reflected in the observed traffic data.

Three main problems arise in predictions made by models: the model can be overconfident, underconfident, or both. Overconfident means that the model overestimates the probability of an event occurring, underconfident means the opposite, and in the case of both the model has both cases at the same time (Baladram, (2025)). Model calibration resolves these issues by altering the input values for the predicted variable until the prediction is in line with the observed situation.

Different calibration measures have been developed in order to get insight into how well a model is calibrated, both visual and mathematical. For the Fietsmonitor, the focus will lie on mathematical calibration measures.

A popular mathematical measure of how well the model is calibrated is the Brier score. This is calculated by the mean squared error (MSE): the squared difference between the predictions made by the model and the observed outcome, divided by the number of observations, as shown in Equation 1.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(p_i - y_i)^2 \quad (1)$$

Here, N is the number of observations, p_i is the model's predicted value of event i, and y_i denotes the observed value of event i. The lower this score is, the better the model is calibrated (Baladram, (2025)).

A similar, but different, mathematical measure is the mean absolute error (MAE). There are a number of benefits to using the mean absolute error compared to the mean squared error. The MAE is more robust to outliers compared to the mean squared error, and is also easier to interpret: the MAE is of the same order of magnitude as the predicted and the observed data, which makes its value more intuitive (Ahmed, (2023)). Equation 2 shows the equation used for calculating the mean absolute error.

$$MAE = \frac{\sum_{i=1}^{N}|p_i - y_i|}{N} \quad (2)$$

Here, p_i represents the predicted value of event i, y_i represents the observed value of event i, and N represents the total number of observations.

To allow for easy interpretation of the results of the model and to lower the impact of outliers, the mean absolute error (MAE) is used to indicate how well the model is calibrated.

3.2 Machine learning and gradient descent

As introduced in section 1.1, the Fietsmonitor multiplies the probability of choosing a route by the traffic in the OD-matrix to obtain the bicycle traffic flow on the network. In order to calibrate this assignment method together with the input data, a stepwise approach can be taken through a machine learning process. Machine learning is a practice where a model improves itself through a programmed algorithm (Banoula, (2025)).

A popular learning algorithm is gradient descent. This algorithm compares one observed value to one predicted value at a time and minimizes a loss function to obtain the best outcome. Loss is defined as the difference between the observed and the predicted value, and is a measure of how well the model is calibrated. The algorithm below explains how gradient descent works.

Algorithm: Gradient descent (Kwiatkowski, (2024))
Initializing: Choose initial parameters (starting point for the calibration)
For k=1 up to K (maximum number of iterations), **do:**

1. Calculate the gradient of the loss function at point k.
2. Scale the parameters being calibrated in the opposite direction of the largest gradient.
3. Repeat until K iterations are finished or the stop condition is met.

Gradient descent is a relatively straightforward and efficient algorithm to implement and use compared to other optimization algorithms. During each iteration, the parameter values are updated in the direction of the steepest descent to enhance the predictions made by the model. In order for the model to converge correctly to the minimum of the loss function, the loss function needs to be convex. In the case of the Fietsmonitor, the loss function is the mean squared error (Equation 1), a mathematical calibration indicator used here to improve the traffic predictions.

A function is convex if the line connecting any two points in the domain lies above the function. Figure 3 visualizes this through a convex and a nonconvex function. As can be seen in the figure on the left, the line connecting any two points within the domain lies above the function. As can be seen in the right figure, the line segment connecting the local minimum and the global minimum lies below the function (Verma, (2023)).

Convexity is an important property for the calibration process: gradient descent uses the gradient of the loss function to update its parameters towards the global minimum of the function. Since the loss is defined as the difference between the observed and the predicted value, the minimum value of the loss function is where the parameters reach their optimal (true) values. Essentially, a convex function

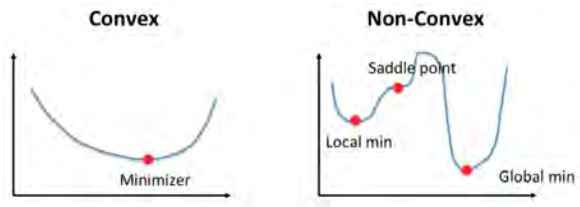

Figure 3: Visualization of a convex (left) and a nonconvex (right) function (Verma, (2023))

only has one minimum value: the global minimum that the model attempts to reach through gradient descent. A nonconvex function may have a local minimum in which the calibration process could get stuck, thinking it has reached the global minimum.

During each iteration the parameters in gradient descent are updated as in Equation 3, where θ denotes the parameters that need to be optimized through the training process, η denotes the chosen learning rate and ∇L is the gradient of the loss function (Zhang et al., (*n.d.*)).

$$\theta^{(i+1)} = \theta^{(i)} - \eta \nabla L \tag{3}$$

The learning rate η of the algorithm indicates the step size during each iteration. Here, there is an important trade-off between convergence rate and accuracy, as shown in Figure 4. A large learning rate may seem feasible for faster convergence, but as shown in the left figure, it may hinder the calibration process instead. A very small learning rate may be more accurate, but it will take a large number of iterations to reach the global minimum, as shown in the figure on the right.

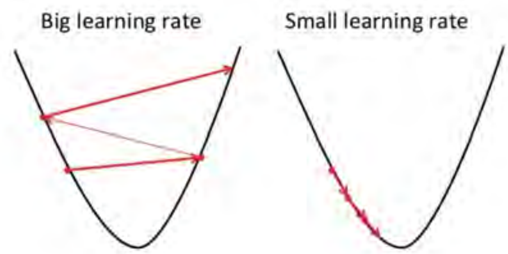

Figure 4: Large vs small learning rate (Mishra, (2023))

As seen in Equation 3, the learning rate η is not calculated in the model, but is a previously determined value. The learning rate is a so-called hyperparameter. These are parameters whose values have been determined before the training process starts (Nyuytiymbiy, (2025)). In the case of the calibration tool for the Fietsmonitor, the only other relevant hyperparameters are the number of iterations and the stop condition.

The stop condition and the number of iterations are important to determine how long the model is allowed to run for. If the stop condition is met, the model will instantly stop its calibration and yield its output. In some cases, reaching this stop condition may take an unfeasible amount of time, or it is maybe not even be possible given the circumstances of the tool. In that case, the maximum number of iterations can be useful in order to end the calibration process.

4 Methodology

Based on the gathered knowledge and understanding of the problem from the previous chapters, a calibration model can be developed. This model uses the theory on machine learning from the theoretical background in section 3 in order to develop a calibration tool that fits the problem statement discussed in section 2. This chapter also discusses the validation process of the available count data and the sensitivity analysis to be conducted on the input data.

4.1 The calibration model

In order for the calibration tool to be developed in Python, four real-life input variables are needed: a geographical network of the study area, OD-matrices, the routes from each origin to each destination, and count data of the city. The model also takes some initial parameters, which can be chosen by the model developer: the method of initialization of the probabilities, the normalization function of the model, the learning rate of the model, and the stop conditions: the maximum number of iterations and the MAE stop condition. In this section, these inputs will be discussed, together with the development process of the model.

4.1.1 Real-life input variables

The infrastructural network that will be used in the calibration process is simplified. For each link (road segment) in the network, it is determined if one has a choice at the end or not. If a road user can only go to one other link when traveling on a road segment, these links are combined into one in order to make the network smaller and easier to interpret for the computer. Besides this, the network is an accurate representation of the study area.

The Fietsmonitor uses OD-matrices as input data, but not all bicycle traffic of all citizens is reflected in this matrix. Hence, the average traffic per person per day is multiplied by the number of citizens in a zone to get an indication of how often a certain trip is made. This gives a good indication overall of the amount of traffic, but is not perfect. Hence, the OD-matrix is not entirely representative of the bicycle traffic in Apeldoorn. To account for this uncertainty, the model should also calibrate the OD-matrix.

The routes constructed in the Fietsmonitor also carry some uncertainty. Only three routes have been constructed by the Dijkstra algorithm. However, cyclists may have more than three routes to choose from when traveling. Besides this, the Dijkstra algorithm calculates the optimal route only in terms of absolute travel distance. This is not entirely realistic: a slightly longer route with fewer turns or traffic signals could potentially have a shorter travel time than these routes, and because of that be more attractive to cyclists.

A lot of count data of cyclists has been made available for the city of Apeldoorn. This can be connected to the infrastructural network in Python to compare it to the predicted loads by the model. The available data is noisy and not entirely reliable. Some locations have far more data counts available compared to others, some locations may only have data available at certain times of the year, or the data may be impacted by external factors. Because of this, it is important to gain insight into what the input data looks like before it can be used in the model. This can be done through a data validation process, as elaborated on in section 4.2.

4.1.2 Chosen input variables

Besides the input variables taken from the real situation, some input variables may also be chosen by the user of the model. These choices are discussed in this section.

The first choice that can be made is in the initialization of the probabilities of choosing each route. There are three possibilities here: initiate with all zeros (each route is chosen with equal probability), entirely random values, or the increment values that are also used to create the routes (0.5, 0.3, 0.2). If the calibration works correctly, the calibration process should always be able to achieve the same mean absolute error. However, the initiation type may have a significant impact on the runtime, as the initial parameters may start at a better starting point and thus converge faster.

Two different normalization functions have been identified for the calibration tool, both with their own advantages. These normalization functions ensure that the updated probabilities of choosing the different routes from origin to destination always add up to one. The first is a standard normalization function, where the probability of choosing each route is divided by the sum of the probability of choosing any route.

$$standard\_normalization(x_i) = \frac{x_i}{\sum_k x_k} \tag{4}$$

The other normalization function is the Softmax function, as shown in Equation 5. The Softmax has several benefits compared to the standard, simple normalization function. The Softmax is able to account for negative values of x_i due to the exponent, and is more robust to large differences between the probabilities (Gökmen and Martinez, (2025)).

$$softmax\_normalization(x_i) = \frac{e^{x_i}}{\sum_k e^{x_k}} \tag{5}$$

The next chosen value is the learning rate of the model. The learning rate determines the step size during each iteration. As mentioned in section 3, it is important that the learning rate is neither too large nor too small.

Lastly, the number of iterations and the stop condition may be manually chosen. The stop condition is the value that the mean absolute error should achieve in order to stop the calibration process. Since the count data and OD-matrices are not entirely reliable, and not all possible routes from origin to destination are created, the mean absolute error does not necessarily need to achieve a value of zero for the calibration to be sufficient. The stop condition can be chosen by the model user and depends on the purpose of the model.

The number of iterations is a nice addition to have alongside the stop condition. For example, if the model does not converge to anything in a run, or if it converges too slowly, the model stops the calibration process when the maximum number of iterations is reached to avoid endless iterations that hardly improve the model. Since the model will be tested on different aspects and the output needs to be compared, for the testing phase, all tests will be conducted for 45 iterations.

During the development of the model, the default values for these chosen parameters will be determined. The chosen number of iterations depends on the rate of calibration of the model. The number should be large enough that conclusions can be drawn from it, but not so large that running the model takes an unfeasible amount of time during the testing process.

Using this outcome, the optimal initiations for the probabilities, normalization function, and learning rate can be found. First, the best normalization function is found through two separate tests: the Softmax normalization function and the standard normalization function. After this, the best probability initiation is found by comparing zeros, increments, and random input. Lastly, the best learning rate is found by testing the efficiency of four different learning rates: 1, 0.5, 0.1, and 0.05.

4.1.3 Model development

Figure 5 (which can be found at the end of this chapter) gives an overview of how the model is going to work. Starting from the top, the main functionalities of the model are explained through the workflow.

Initially, some inputs are needed, such as the OD-matrix and the count data. It also needs the infrastructural network of the study area. This information can then be used to start the calibration tool. Before the calibration iterations start, the initial traffic flow on the network and the reliability factors of the different count locations need to be calculated, as well as the count data.

Using all these input elements, the calibration iterations can start. Four different kinds of calibration take place in the model:

- **Probabilities update:** this function updates the probability of choosing the shortest, second shortest, or third shortest route from each origin to each destination.
- **OD-matrix update per route:** this function updates the traffic in the OD-matrix per origin-destination pair.
- **Overall OD-matrix update:** this function updates the entire OD-matrix by a factor that represents the average over- or underestimation of predicted traffic compared to the observed traffic.
- **Count data update:** this function updates the count data at the different locations based on the gap between the predictions and observations.

4.2 Data validation

As mentioned previously, the count data made available for this project is not entirely reliable. The amount of data available differs largely between the locations, and at some locations, the data is outdated. Hence, this data needs to be analyzed and validated. The reliability of the count locations is translated into the reliability factor used in the model to update the probabilities and OD-matrix during each iteration. This means that more reliable count locations will have more impact on the calibration process of the model compared to less reliable ones. The count data is given a reliability factor based on the following information:

1. **Amount of count data available:** if there is a large amount of data available over a large number of years, it can be said that the data is relatively reliable.
2. **Last available count data:** a count location with its most recent count data in 2024 is more reliable compared to a location with its latest count in 2021.
3. **Standard deviation in the count data:** if there are large fluctuations between the days in the dataset of a count location, this might make the data less reliable. Some fluctuations in day-to-day traffic are natural, of course, but a standard deviation of (for example) 500 indicates that something is wrong with the dataset.
4. **Amount of routes passing a count location:** if only one route passes a count location, it can be safely assumed that this data point says more about the traffic on this route than another data point that has 20 other routes passing by.

Depending on the outcome of all these characteristics, the reliability factor is calculated for each count location. The reliability factor is the product of multiplying the scores given to a location based on the above-mentioned characteristics. The scores are given depending on how well that characteristic of the count location performs compared to the other data.

The scores of the first three criteria are shown in table 1. In addition to these criteria, the number of routes passing a count location is also considered for the reliability factor. Locations that are passed by a large number of routes get a lower score compared to locations that are only passed by a small number of routes. This is because a small number of routes passing means that the count location says more about that specific route. For example, if only one route passes a location, the observed traffic in that location must all come from that route. How this score is calculated is shown in Equation 6, where x_i is the number of routes passing count location i, and $\sum x$ denotes the summed amount of routes passing all count locations.

$$Score = \frac{(\sum x) - x_i}{\sum x} \tag{6}$$

Procentual standard deviation	>1.2	>0.9	>0.6	>0.4	<0.4
Score	0.5	0.7	0.8	0.9	1
Number of counts	<500	<1000	<7000	<15000	>15000
Score	0.5	0.7	0.8	0.9	1
Last available count	<01-01-2021	<01-01-2022	<01-01-2023	<01-01-2023	
Score	0.4	0.6	0.8	1	

Table 1: Reliability score of different data characteristics

4.3 Sensitivity analysis

Next to an assessment on the reliability of the data, an assessment on the sensitivity of the model should be made. If the model is very sensitive to changes in the input data, it is less reliable than a more robust model that always converges to the same outcome.

The model will, as mentioned in section 4.1.3, be tested on different initiation parameters to find the optimal outcome. This outcome is already a good indicator of the robustness of the model: changing the chosen input variables should not impact the calibration process of the model; it should only impact the number of iterations it takes the model to converge.

The reliability scores of the count data characteristics are also merely an estimate of how (un)reliable the characteristics make the count location. Therefore, seeing the impact it has on the model is a useful addition. The scores will be altered a number of times to see what the impact on the outcome of the model and the calibration process is.

Lastly, the initial OD-matrix is also not a perfect reflection of reality. The current OD-matrix estimates the total amount of traffic by multiplying the reported traffic by the average amount of traffic per person per day. However, this is only one way to estimate the OD-matrix, and there may be other, better methods to estimate the true OD-matrix. These methods are outside the scope of this thesis. To test the impact the OD-matrix has on the outcome of the model, the calibration tool will be tested with an alternative OD-matrix.

This alternative OD-matrix classifies the distance of all OD-pairs within 10 distance classes (between 0.5 and 10 kilometers). The OD-matrix then takes the mean number of trips in each distance class and replaces the traffic in each OD-pair with this.

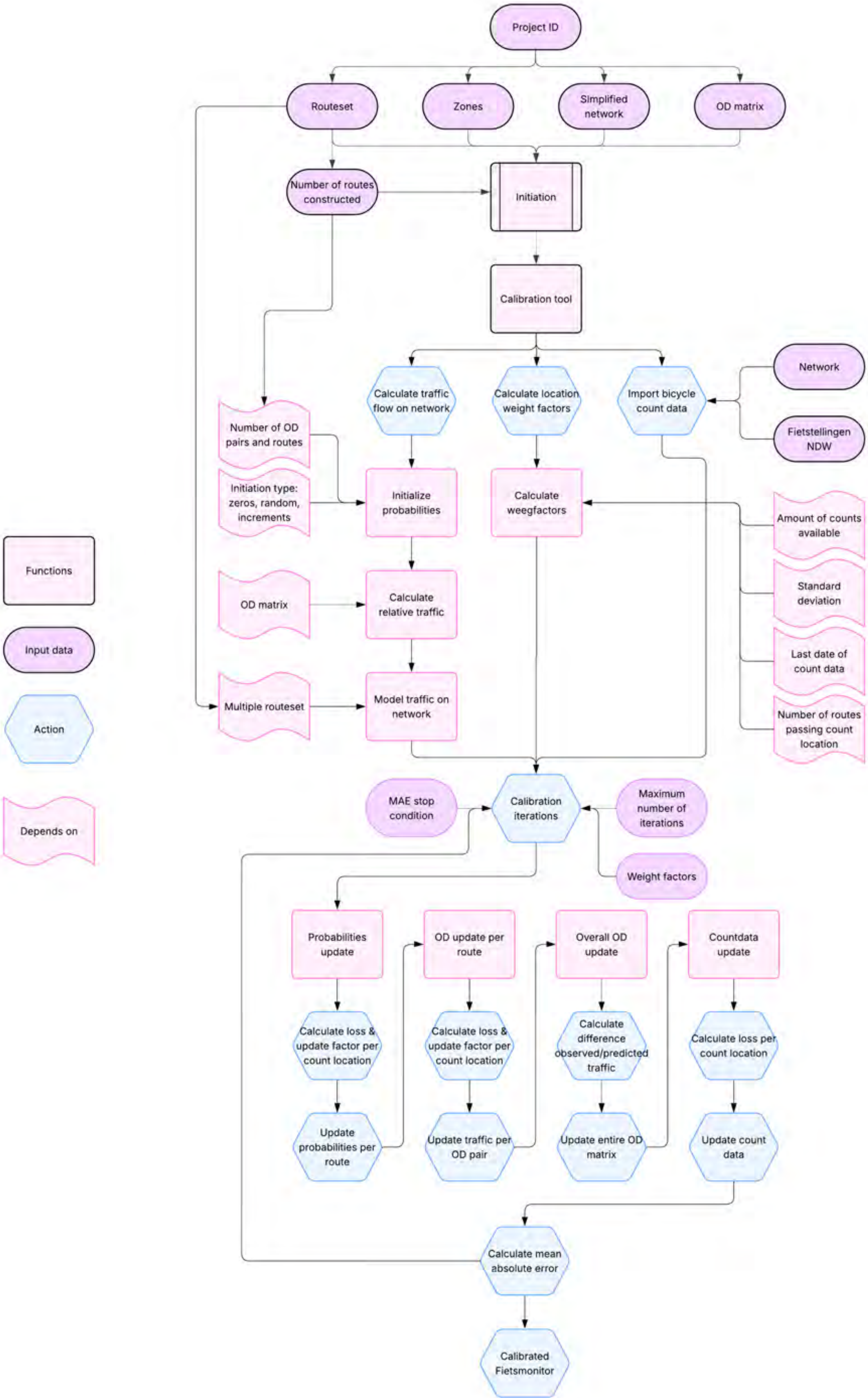

Figure 5: Overview of the calibration model structure

5 Results

Using the methodology from section 4, the Python calibration tool could be constructed. In section 5.1, the initial outcome of the model is presented before optimizing the chosen input variables. After this, section 5.2 finds the optimal initial settings for the calibration tool. Once these have been established, section 5.3 tests the sensitivity of the model. Lastly, section 5.4 shows the optimal outcome of the Fietsmonitor.

5.1 Initial results

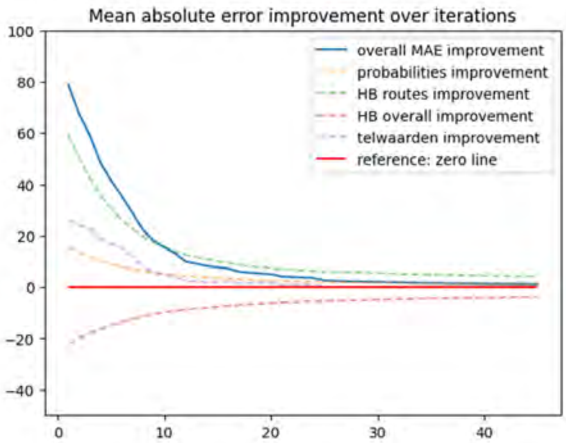

Figure 6: Initial output of the model (original case)

The 'probability' calibration improvement depends on the model altering the probability of choosing the shortest, second shortest, or third shortest route in each OD-pair. Its improvement depends on both the initialized values of the probabilities, the normalization function used for the probabilities, and the learning rate of the model. All of these characteristics will be assessed in section 5.2.

'HB routes' (OD-routes) calibration allows the model to update the traffic within each origin-destination pair to be more in line with the observed amount of traffic. This type of calibration is necessary to account for the degree of uncertainty in the OD-matrix provided.

'HB overall' (OD-overall) calibration is an interesting type of improvement, as it is the only type of calibration that yields negative improvement (hence: making the model worse). HB overall improvement looks at the total amount of observed traffic and the total amount of predicted traffic, and updates the entire OD-matrix to be closer to the amount of observed traffic. This prevents the model from overfitting the data to the OD-pairs by generating an endless amount of traffic in order to fit the observed traffic to the observations.

The calibration of the 'telwaarden' (count values) is necessary to account for the uncertainty of the count data. Since allowing the model to endlessly alter the count values is also unreasonable and runs the risk of overfitting, the model updates the count values until the loss (gap between observed and predicted traffic at a location) is within the interval presented in Equation 7. There are large gaps between the observed traffic: some locations only observed 100 bicycles, whereas other locations have observed 5000. Hence, it is necessary to have a variable loss interval rather than a set standard value.

$$-100 - \left(\frac{1}{4} * observed\_traffic\right) < Loss < 100 + \left(\frac{1}{4} * observed\_traffic\right) \tag{7}$$

5.2 Optimizing chosen variables

Now that the initial results have been established, it is now the goal to find the optimal characteristics for the model to obtain the best mean absolute error (MAE) in the smallest number of iterations. The MAE denotes the average amount of bicycles that the model over- or underestimates compared to the observed count data. This is done by reviewing the type of normalization function used, the initialized probabilities, and the learning rate.

5.2.1 Normalization function

To start, the best normalization function needs to be chosen. Two options have been determined for the normalization function: the Softmax and the standard normalization function. Both of these functions ensure that the probabilities of choosing the different routes from origin to destination add up to one, however, the normalization function does have a large impact on the convergence time of the MAE. The tool was tested twice, using a learning rate of 0.1 and a probability initiation as 'increments' for testing purposes. The outcome of these tests is shown in appendix A.1 and Figure 7 and Figure 8.

As can be seen from these figures, the improvement of the MAE is faster using the standard normalization function. At the end of the 45 iterations, the calibration using the Softmax function reaches an MAE of 163.82 with a percentage error of 21%. The standard normalization function reaches an MAE of only 93.68, which is equal to a percentage error of only 12%. Therefore can be seen that the standard normalization function is preferable in the calibration tool, and will be used for further testing.

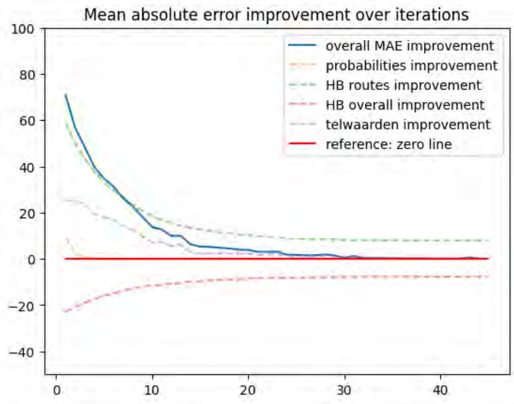

Figure 7: Softmax normalization

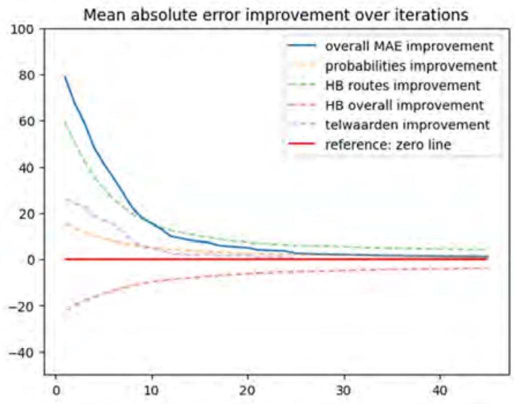

Figure 8: Standard normalization

5.2.2 Initial probabilities

Now that the normalization function has been decided upon, the probabilities initialization type can be decided upon. Here, three different options have been identified: initializing all probabilities as zeros (thus, the probability of choosing each route is equal), as increments (previously determined as 0.5, 0.3, 0.2), and lastly as random values.

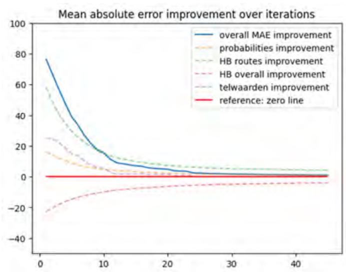

Figure 9: Probabilities initialized as zeros

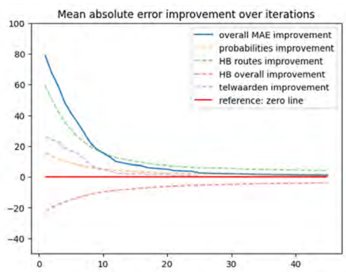

Figure 10: Probabilities initialized as increments

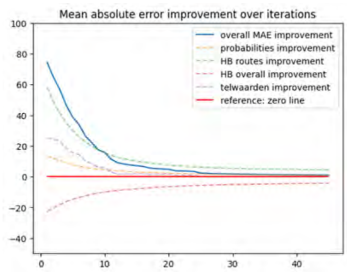

Figure 11: Probabilities initialized as random

The outcome of these tests can be found in appendix A.2, as well as in Figure 9, Figure 10, and Figure 11. As expected, the model still converges similarly, as the only difference is the start value of the MAE. Due to the gradient descent, a worse starting value results in a steeper descent at the start, but eventually all models should reach a similar end value.

Since the main difference is the starting point and not necessarily the speed of convergence, one type of initiation has the lowest MAE at the end of the 45 iterations. The random initiation was revealed to be the worst starting point, having an end value for the MAE of 98.39, and an error of 12%. The MAE of the probabilities initialized as increments was the lowest, with only 93.68 and an error of 12%

Initiating the probabilities as all zeros performed only slightly worse than the increments, with an end value of the MAE of 93.84. The model using the initiation type 'zeros' had to change significantly less about the OD-matrix in order to obtain this value for the MAE. To preserve the original OD-matrix to the best extent, it was decided to continue working with the probabilities initialized as zeros.

5.2.3 Learning rate

Four different values for the learning rate have been evaluated. These outcomes can be found in appendix A.3, as well as Figure 12, Figure 13, Figure 14 and Figure 15. The issue noted in section 3.2 on a learning rate that is too large is perfectly represented in Figure 12: at first, the model seems to perform really well, but after only a few iterations, the model is unable to converge to anything. It can also be seen that a learning rate that is quite small, such as 0.05 (Figure 15), significantly slows down the calibration process.

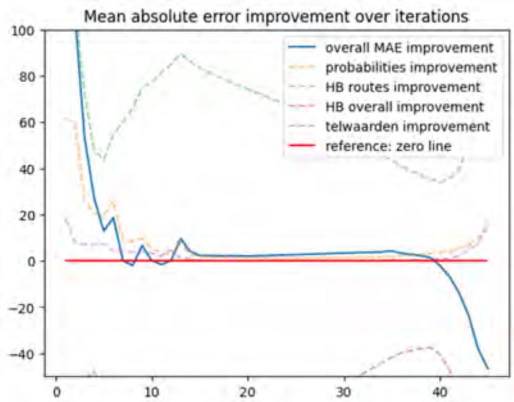

Figure 12: MAE improvement over iterations, learning rate 1.0

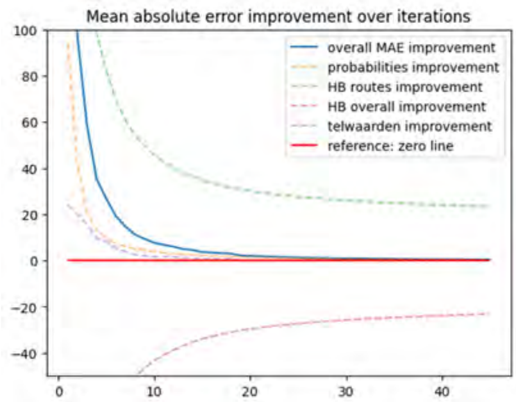

Figure 13: MAE over iterations, learning rate 0.5

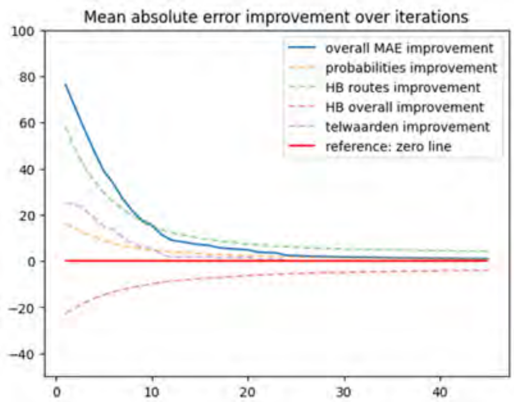

Figure 14: MAE improvement over iterations, learning rate 0.1

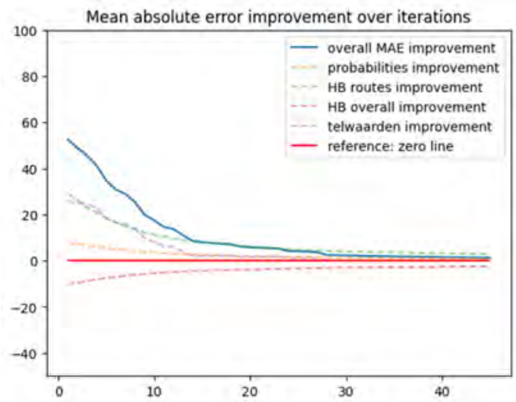

Figure 15: MAE over iterations, learning rate 0.05

Learning rates of 0.5 and 0.1 both seem to perform quite well compared to the other two learning rates. With a final MAE value for the learning rate of 0.5 at 90.47, and 0.1 at 93.84. The learning rate of 1.0 had an outcome of an MAE of 217.90, and the learning rate of 0.05 had an outcome of an

MAE of 111.6.

To analyze the impact of the learning rate over a larger number of iterations, the learning rates of 0.5, 0.1, and 0.05 were also tested over 100 iterations. From the current output can already be concluded that a learning rate of 1.0 is insufficient, and thus this learning rate does not need to be tested again. From these tests was found that the learning rate of 0.5 ends the last iteration with an MAE of 81.16. Unexpectedly, the smallest learning rate (0.05) performs better: ending at 74.87. The learning rate of 0.1 yields the best result, having an MAE of only 64.21 at the end of the 100 iterations. The best learning rate to use over a large number of iterations has been identified as 0.1.

5.3 Sensitivity analysis

Now that the initial parameters have all been given their optimal initiation, the sensitivity of the model to changes in the input data can be found. This is done by testing the reaction of the model to a different OD-matrix and to different values for the reliability factors.

5.3.1 OD-matrix sensitivity

To review the impact the OD-matrix has on the calibration of the model, an alternative OD-matrix was constructed, as explained in section 4.3. When comparing Figure 16 to Figure 14, it can be seen that these improvement graphs behave similarly. The model using the new OD-matrix seems to converge faster in the beginning. However, this is a natural result of having a worse starting point (and thus will lead to a steeper descent).

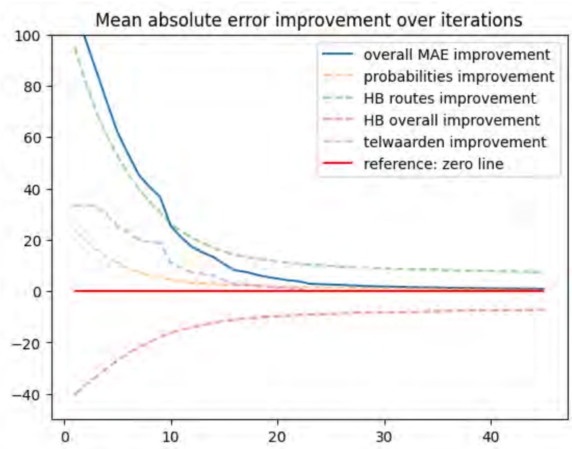

Figure 16: MAE improvement over iterations, different OD-matrix

As can be seen in the figure, the model starts off at a worse value for the MAE, but behaves similarly to the standard OD-matrix over iterations. Even though the end result of the model after 45 iterations may be worse, the similarity in the calibration process implies that a similar result can be achieved if given more processing time.

5.3.2 Reliability factor sensitivity

Before discussing the reaction of the mean absolute error to changes in the reliability factors, it should be noted that the MAE also depends on the reliability factors. Locations with a low reliability factor have a lower share in the MAE compared to locations with a high reliability factor. Therefore, only the difference in improvement rate of the MAE using a different reliability factor can be compared.

Comparing Figure 18 and Figure 19 to Figure 17, it can be seen that the calibration process performs similarly to the model using the original reliability factor. The difference in the reliability factors makes for a different starting value of the MAE, and also impacts the steepness of the graph. However, the model still behaves similarly and does not make any weird or unexpected calibration steps.

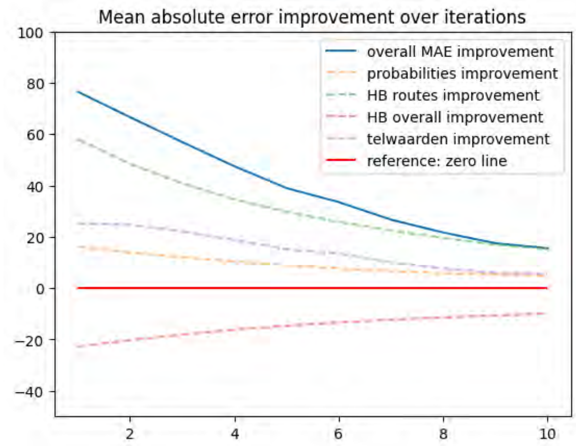

Figure 17: MAE improvement over 10 iterations, standard reliability factors

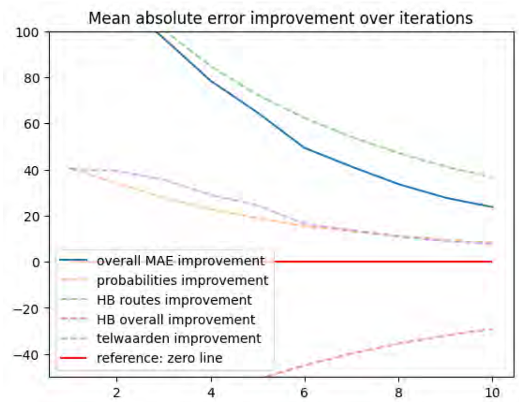

Figure 18: MAE improvement over 10 iterations, all reliability factors equal 1

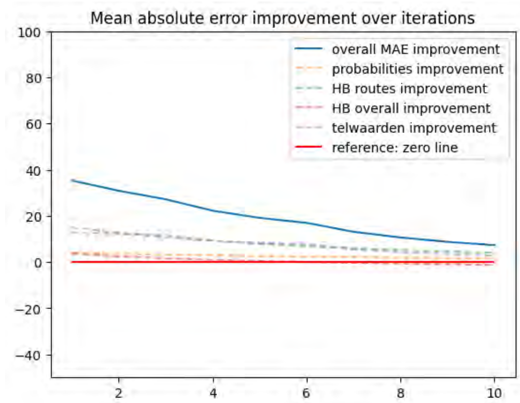

Figure 19: MAE improvement over 10 iterations, all reliability factors halved

5.4 Optimal model

After obtaining the results on the best initial settings for the model, the model was tested for 100 iterations. All four different calibration types work together to obtain this result, and are all impacted by the initial settings. The model starts off with an MAE of 610.76. At the end of the 100 iterations, the model was calibrated to an MAE of 64.21, which is a percentage error of only 8.5%. The improvement process of the optimal model is shown in Figure 20.

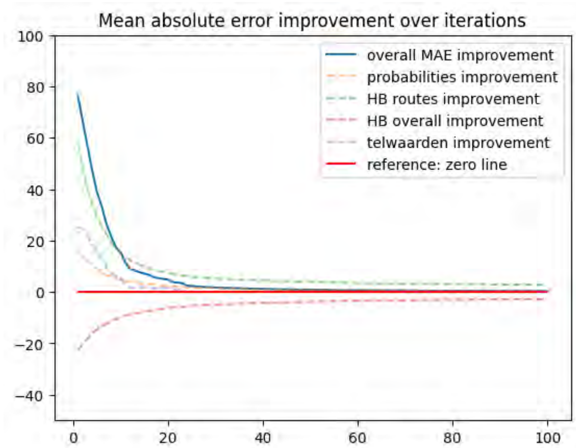

Figure 20: Optimal MAE calibration

This calibration can be visualized in QGIS (Quantum Geographic Information System), as can be seen in Figure 21 and Figure 22. In both figures, the predictions made by the model are presented as a traffic flow over the network. The wider and yellower a road segment is, the larger the predicted traffic load is. The original bicycle traffic predictions are presented in Figure 21, whereas Figure 22 shows the calibrated Fietsmonitor. It can already be seen that the calibrated model is a lot more nuanced and has fewer segments overflowing with traffic. But also when investigating the count data from the city center, where most bicycles travel to and from, the calibrated model is far more in line with the count data. Larger versions of these figures can be found in appendix C.1.

Figure 21: Original bicycle traffic predictions

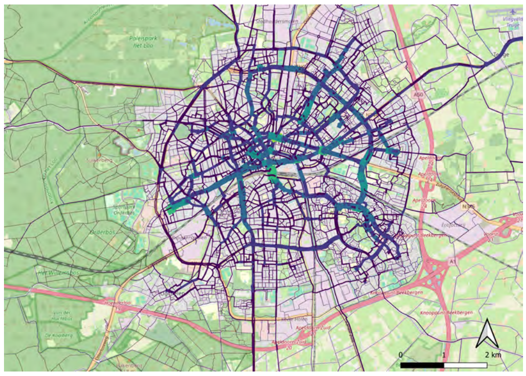

Figure 22: Calibrated bicycle traffic predictions

Figure 23 shows the original count data from NDW, using the same scale as Figure 21 and Figure 22. Figure 24 presents the calibrated count data. As can be seen, the corrections made by the model to the count data are not substantial. Larger versions of these figures can be found in appendix C.2.

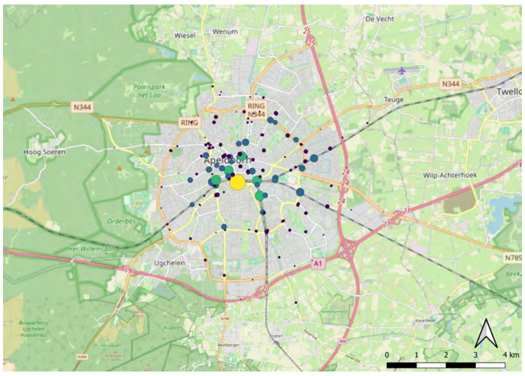

Figure 23: Original count data

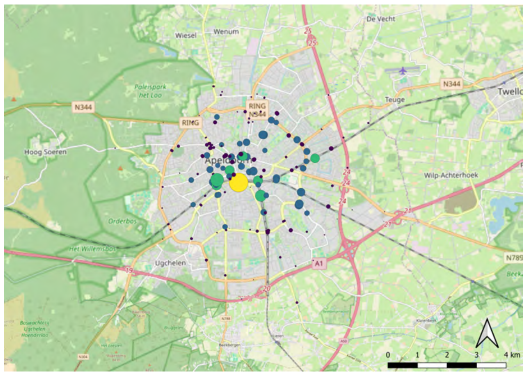

Figure 24: Calibrated count data

6 Discussion

From the results in section 5 can be concluded that a machine learning calibration approach is a valuable addition to the current state of the Fietsmonitor. It allows for better bicycle traffic predictions that are more in line with the observed traffic, while also accounting for the uncertainty in the input data of the model.

It was found that a standard normalization performs better compared to the Softmax function. This is most likely due to the larger differences in the normalized probabilities. The Softmax function not only normalizes the probabilities but also accounts for outliers and brings all probabilities closer together. The standard normalization function does not do this, meaning the calibration of the probabilities has more impact on the model, and thus converges faster to the goal value.

A somewhat surprising result was the initial probabilities outcome. The three routes from each origin to each destination have been constructed with the following increments: 0.5, 0.3, 0.2. This implies it would seem logical for this to reflect in the initial probabilities. However, it was found that the difference in the MAE was negligible between the increments and the initiation type 'zeros'. Next to the MAE, it was found that the initial probability type 'zeros' needed to alter the OD-matrix substantially less in order to arrive at this result. Because of this, the 'zeros' initiation type was found to be the best type to use.

Reviewing the result of the relative probabilities yielded another surprising result. If all OD-pairs that do not pass a count location (and thus have not been calibrated) are excluded, the third shortest route has the highest probability of being chosen on average. The shortest route has an average probability of being chosen of 30.2%, the second shortest route has an average probability of 30.3%, and the third shortest route has an average probability of being chosen of 39.6%. This is an unexpected outcome of the model, as one would expect the shortest route to be the most feasible one. One explanation of this has already been mentioned in section 4.1.1: the shortest route may be the shortest in terms of absolute distance, but this does not necessarily mean it is an attractive route to travel by.

This result also comes with another issue: the route set has been constructed using the initial probabilities type 'increments'. Now that it has been found that this is not the optimal initiation method, it implies an extra uncertainty within the developed route set. In future route set developments, it may be useful to construct a route set with equal probabilities between all three of the routes, or the relative probabilities found by the model.

In the last test, the optimal learning rate was identified as 0.1. This was also the initial value used, meaning it aligns with the predicted outcome. It was found that a larger learning rate may seem better during the first iterations, but the learning rate 0.1 outperformed a larger learning rate over a larger number of iterations. The cause of this can be seen in Figure 4.

In order for the model to calibrate, substantial changes needed to be made in the traffic in the OD-matrix. The initial OD-matrix has a total of 290579.1 trips stored in it. After 100 iterations, the calibrated OD-matrix differs by 69958.94 trips from the original one. This means that almost a quarter of the trips have been taken out of the OD-matrix. This may be due to the OD-routes calibration process in combination with the OD-overall calibration process. The OD-routes calibration calibrates only the OD-pairs that pass a count location in the network. The calibration of the overall OD-matrix updates the entire matrix based on a factor calculated by comparing all observed traffic to the respective predicted traffic. If the OD-routes calibration constantly creates traffic, the overall OD-matrix calibration constantly decreases the number of trips in the overall OD-matrix. This means that in each iteration, trips are taken away from the OD-pairs that do not pass any count locations.

While the model seems to perform well in Apeldoorn, there are some limitations to it. For example, the model can only calibrate on routes that pass a count location where traffic has been observed. In total, 1048576 OD-pairs are in the OD-matrix in Apeldoorn. From all these OD-pairs, only 56078 have at least one route passing at least one count location. Here should be noted that there are also many OD-pairs from an origin to a very close destination, which gives the route only a small chance to pass a count location. Having more count data available at more locations throughout the study area could resolve this issue.

Another limitation is the number of routes constructed in the Fietsmonitor. Only three routes are constructed, even though people may travel over a wide variety of routes if given the choice. This means that the model sends all traffic over the constructed routes, and no traffic over the other possible routes. This would mean that the model can never truly reflect the real situation. This issue could be resolved if more routes were constructed initially. In this case, this would also raise the question of how many routes would be needed. A tradeoff would need to be found between the precision of more routes and the increased processing time of the calibration tool.

These limitations do not form an issue for the purpose of the model. The goal and use of the model would not necessarily require the outcome to be exact. The calibrated model reflects the reality far better than the uncalibrated model, shown in the decrease in the MAE from 610.76 to 64.21. This means that the calibration tool can already fulfill its purpose: provide improved, substantiated arguments for bicycle infrastructure design.

All in all, the model has achieved a satisfactory result: better bicycle traffic predictions in Apeldoorn. The calibration tool can be applied to different cities in the Netherlands, as long as there is count data available and some indication of what the OD-matrix looks like.

7 Conclusion

In this study, a machine learning approach has been taken to calibrate the Fietsmonitor. This is done using two methods: the reliability factor and the calibration process. The calibration resolves large gaps between prediction and observation, while also accounting for the uncertainty of the input data.

Of the three traffic assignment methods tested (zeros, increments, and random), all three lead to similar end results. Since the equal ('zeros') assignment method has a satisfactory outcome of the MAE and changes the least in the OD-matrix in this process, this method is preferred.

The calibration model outputs two graphs: showing the overall MAE and the improvement of the MAE due to the different calibration types. Next to this, a detailed visualization of the bicycle traffic load is provided through a geopackage that can be used in QGIS. This can be used to visualize the difference made in the bicycle intensities in the network as compared to the uncalibrated model. The calibration model was optimized using the MAE as the measure of calibration.

As found in the sensitivity analysis, the model responds quite well to changes in the input data. These changes may affect the speed of the convergence of the calibration and the starting point, but the models calibrate in a similar manner. This means that the calibration tool is a reliable model that matches the sensitivity demands to compensate for the uncertainty in the input data.

The calibration brought the MAE down significantly from 610.76 to 64.21 in 100 iterations. This means that adding the calibration tool using machine learning with gradient descent to the Fietsmonitor greatly improves the traffic predictions, while maintaining a reasonable processing time. To conclude, the developed calibration tool was found to be a valuable addition to the Fietsmonitor.

8 Recommendations

Using this new calibration approach, a lot more possibilities have opened up for the improvement of the Fietsmonitor. These can either be concerned with the current calibration tool itself or with possible future additions to the tool.

One improvement of the current model would be finding better and more reliable increments, and also using these to create the routeset. As of right now, the model performs best using equal probabilities for each of the routes. However, while previous studies have found that not every cyclist travels over the shortest route, it remains one of the largest decision variables. Therefore, further testing of the initial probabilities could reveal a better initial division.

Calculating a new route set using an equal-probability division could also potentially improve the Fietsmonitor. In the current state of the calibration tool, the construction increments of the route set do not match the initialized probabilities of the traffic. This new route set could improve the outcome of the calibration tool.

An addition that would build onto this is changing the number of routes created by the model. Cyclists may travel over a wide variety of routes if given the possibility, meaning that three routes may be an insufficient amount. Of course, this would have an impact on the processing time of the model, as more calculations would need to be made during each iteration. This would then also raise the question of what the optimal increment division would be for a different number of routes created.

An entirely new probability division could also prove to be a useful addition to the script. One option that may yield interesting results is a probability division based on the route characteristics. In previous literature, many different decision characteristics have been identified that impact the route choice process of cyclists. Using these characteristics and a logistic regression model could, for example, yield a very realistic initial probability division between the different routes. A better initial probability division would result in a lower processing time for the calibration tool.

When reviewing the decision variables, the mode of transport may also be a relevant factor to look into. RIVM, (2022) has found that around 30% of the Dutch population travels by an e-bike. E-bikes allow easier travel with low effort. E-bike users may not care as much about small differences in travel distances compared to other spatial characteristics that provide more comfort.

Another possibility to look into to decrease the calibration time of the model may be a different machine learning algorithm. Gradient descent is a straightforward and well-known algorithm to use, but there exist many more algorithms. A study into these algorithms and seeing if any algorithm provides a better outcome may be very beneficial to the runtime of the model.

Next to different machine learning algorithms, there also exist line search methods. These methods calculate the learning rate of the model during each iteration to find a learning rate that fits the model best at that state of the learning process (Nayak, (2022)). Such an addition to the calibration tool could significantly improve the number of iterations that the model would need to calibrate. However, due to the additional calculations needed during each iteration, it would most likely increase the processing time needed for each iteration. It should therefore be reviewed whether this tradeoff is feasible.

References

Ahmed, M. W. (2023). Understanding mean Absolute error (MAE) in Regression: A Practical guide. https://medium.com/@m.waqar.ahmed/understanding-mean-absolute-error-mae-in-regression-a-practical-guide-26e80ebb97df

Baladram, S. (2025, February). Model Calibration, Explained: A Visual Guide with Code Examples for Beginners. https://towardsdatascience.com/model-calibration-explained-a-visual-guide-with-code-examples-for-beginners-55f368bafe72/#:~:text=Model%20calibration%20measures%20how%20well,of%20its%20predictions%20being%20correct.

Banoula, M. (2025, April). Machine Learning Steps: A complete guide. https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-steps

Centraal Bureau voor de Statistiek. (2024, July). Onderweg in Nederland (ODIN) 2023 - onderzoeks-beschrijving. https://www.cbs.nl/nl-nl/longread/rapportages/2024/onderweg-in-nederland--odin---2023-onderzoeksbeschrijving?onepage=true

Deepchecks. (2022, December). What is Binary Classification | Deepchecks. https://www.deepchecks.com/glossary/binary-classification/

Deepchecks. (2024, May). What is Model Calibration? Methods & When to Use. https://www.deepchecks.com/glossary/model-calibration/

de Haas, M., & Kolkowski, L. (2023, November). *Cycling facts 2023* (tech. rep.). Nederlands institute for transport policy analysis.

Gökmen, B., & Martinez, D. (2025, February). Why use Softmax as opposed to simple normalization. https://www.baeldung.com/cs/softmax-vs-simple-normalization-neural-nets

Harms, L., Bertolini, L., & Brömmelstroet, M. T. (2014). Spatial and social variations in cycling patterns in a mature cycling country exploring differences and trends. *Journal of Transport & Health*, *1*(4), 232–242. https://doi.org/10.1016/j.jth.2014.09.012

Keylabs. (2024, September). Types of Classification Models:Binary, Multiclass&Multilabel | Keylabs. https://keylabs.ai/blog/types-of-classification-models-binary-multiclass-and-multilabel/#:~:text=Binary%20classification%20is%20the%20simplest,to%20have%20multiple%20labels%20simultaneously.

Kwiatkowski, R. (2024). Gradient Descent Algorithm — a deep dive - TDS Archive - Medium. https://medium.com/data-science/gradient-descent-algorithm-a-deep-dive-cf04e8115f21

Łukawska, M., Paulsen, M., Rasmussen, T. K., Jensen, A. F., & Nielsen, O. A. (2023). A joint bicycle route choice model for various cycling frequencies and trip distances based on a large crowdsourced GPS dataset. *Transportation Research Part A Policy and Practice*, *176*, 103834. https://doi.org/10.1016/j.tra.2023.103834

Meister, A., Felder, M., Schmid, B., & Axhausen, K. W. (2023). Route choice modeling for cyclists on urban networks. *Transportation Research Part A Policy and Practice*, *173*, 103723. https://doi.org/10.1016/j.tra.2023.103723

Mishra, M. (2023). The learning rate: a hyperparameter that matters - Mohit Mishra - medium. https://mohitmishra786687.medium.com/the-learning-rate-a-hyperparameter-that-matters-b2f3b68324ab

Mobiliteit.nl. (2024, April). Proef met snelle fietsers op rijbaan van start in Amsterdam - Mobiliteit. https://www.mobiliteit.nl/mobiliteitsbeleid/2024/04/18/proef-met-snelle-fietsers-op-rijbaan-van-start-in-amsterdam/?gdpr=accept

Nayak, A. (2022). Line-Search methods for Optimization - Abhijeet Nayak - Medium. https://medium.com/@abhijeetknayak/line-search-methods-for-optimization-28eacddd95ec

Nyuytiymbiy, K. (2025, January). Parameters and hyperparameters in machine learning and deep learning. https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac/

OpenStreetMap. (n.d.). OpenStreetMap. https://www.openstreetmap.org/#map=13/52.20266/5.99682

RIVM. (2022, November). Almost a third of the cyclists in the Netherlands use electric bikes. https://www.rivm.nl/en/news/almost-third-of-cyclists-in-netherlands-use-electric-bikes

Ton, D., Cats, O., Duives, D., & Hoogendoorn, S. (2017). How Do People Cycle in Amsterdam, Netherlands?: Estimating Cyclists' Route Choice Determinants with GPS Data from an Urban Area. *Transportation Research Record Journal of the Transportation Research Board*, *2662*(1), 75–82. https://doi.org/10.3141/2662-09

Van Nijen, N., Ulak, M. B., Veenstra, S., & Geurs, K. (2024). Exploring factors affecting route choice of cyclists: A novel varying-contiguity spatially lagged exogenous modeling approach. *Journal of Transport and Land Use*, *17*(1), 557–577. https://doi.org/10.5198/jtlu.2024.2452

Veenstra, S. (n.d.). FietsMonitor. https://www.witteveenbos.com/nl/digital-solutions/bicycle-traffic-monitor

Verma, A. (2023). Navigating the Terrain: Convex vs. Non-Convex Functions in Optimization. https://ai.plainenglish.io/navigating-the-terrain-convex-vs-non-convex-functions-in-optimization-86812e9a1989

Wegverkeer, N. D. (n.d.). NDW. https://www.ndw.nu/ndw

Witteveen+Bos. (n.d.). FietsMonitor Haarlem. https://www.witteveenbos.com/nl/projects/fietsmonitor-haarlem

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (n.d.). *Dive Into Deep Learning*. https://www.d2l.ai/chapter_optimization/gd.html

A Appendix A: initial parameter tests

A.1 Normalization function

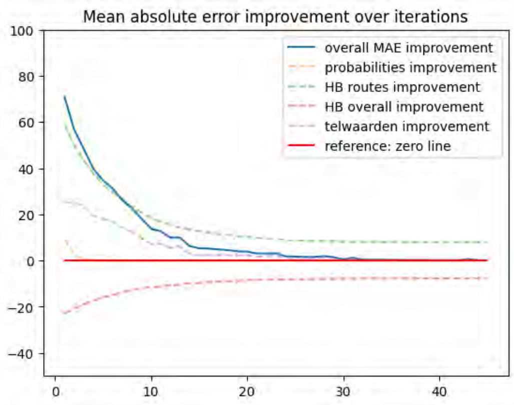

Figure 25: MAE improvement over iterations, Softmax normalization (repetition of figure 7)

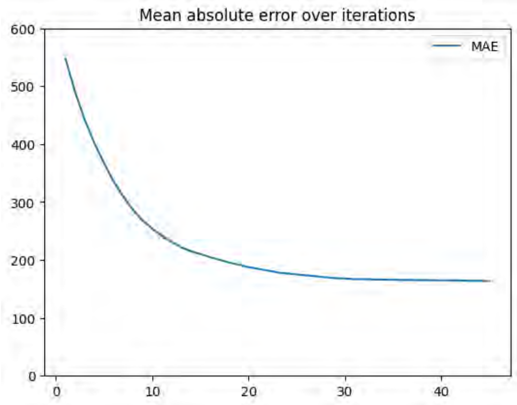

Figure 26: MAE over iterations, Softmax normalization

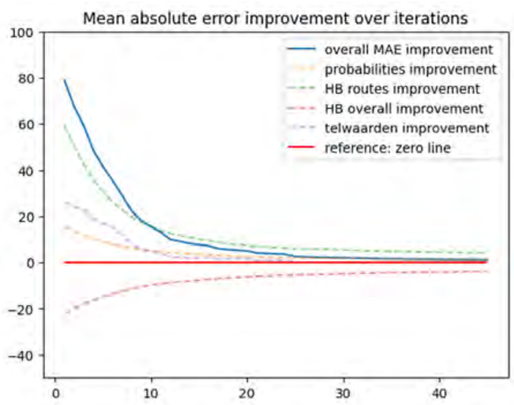

Figure 27: MAE improvement over iterations, standard normalization (repetition of figure 8)

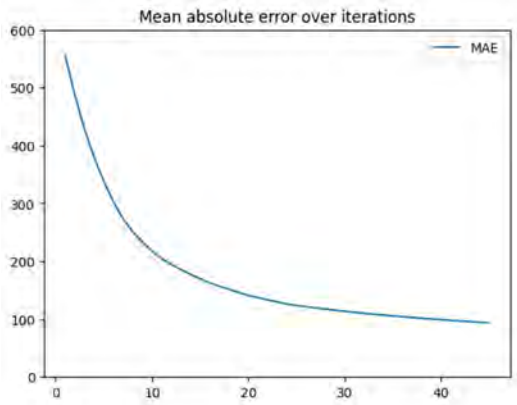

Figure 28: MAE over iterations, standard normalization

A.2 Initiation type probabilities

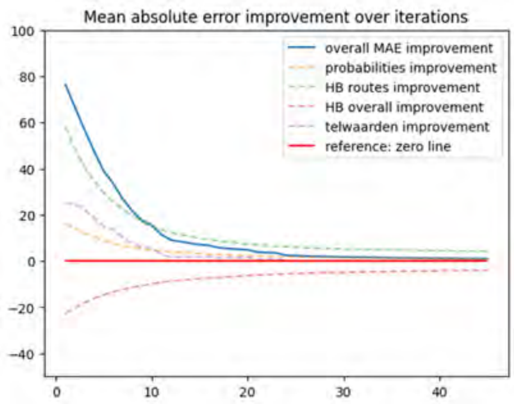

Figure 29: MAE improvement over iterations, initial probabilities zeros (repetition of figure 9)

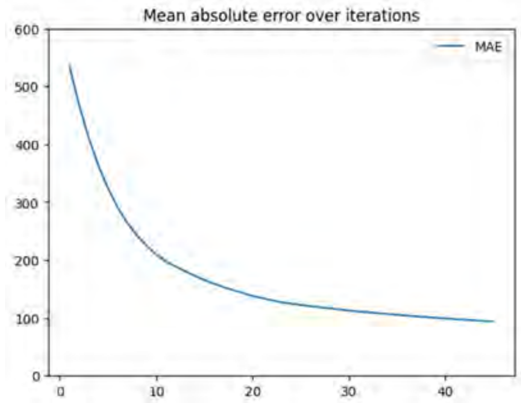

Figure 30: MAE over iterations, initial probabilities zeros

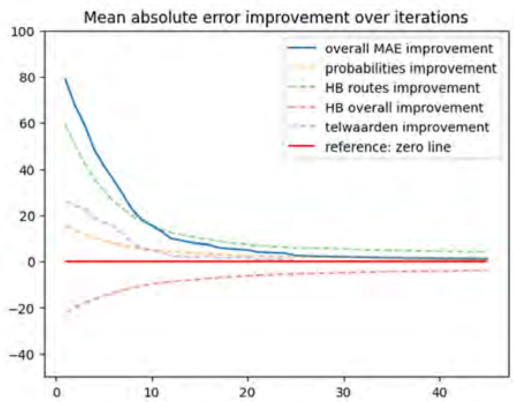

Figure 31: MAE improvement over iterations, initial probabilities increments (repetition of figure 10)

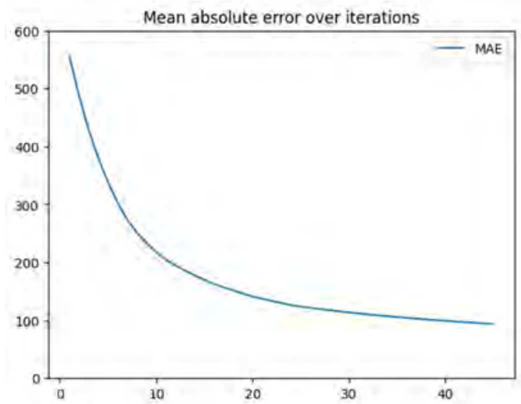

Figure 32: MAE over iterations, initial probabilities increments

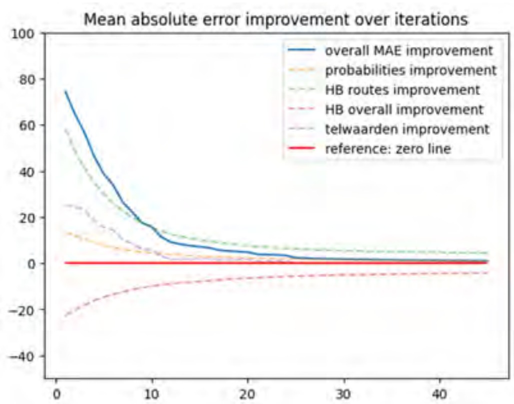

Figure 33: MAE improvement over iterations, initial probabilities random (repetition of figure 11)

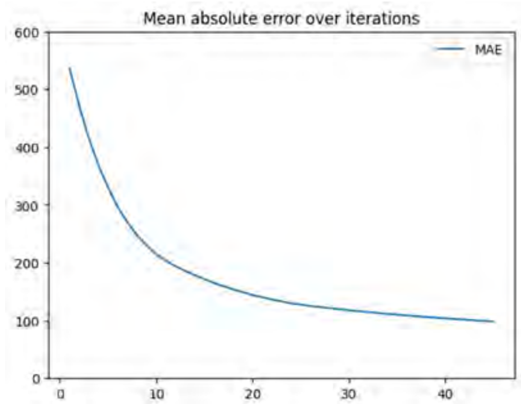

Figure 34: MAE over iterations, initial probabilities random

A.3 Learning rate

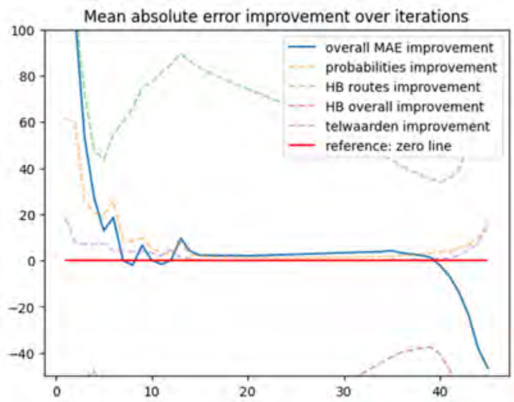

Figure 35: MAE improvement over iterations, learning rate 1.0 (repetition of figure 12)

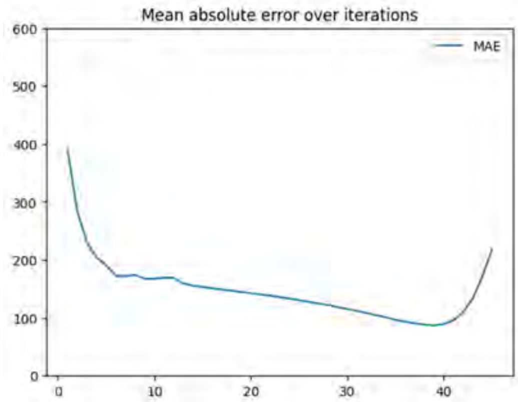

Figure 36: MAE over iterations, learning rate 1.0

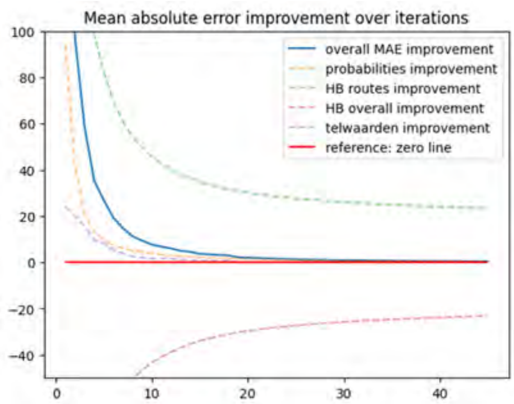

Figure 37: MAE improvement over iterations, learning rate 0.5 (repetition of figure 13)

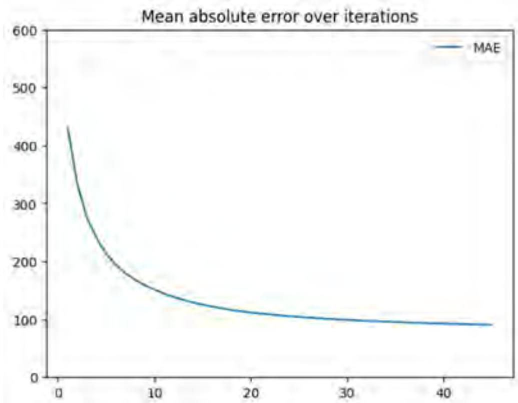

Figure 38: MAE over iterations, learning rate 0.5

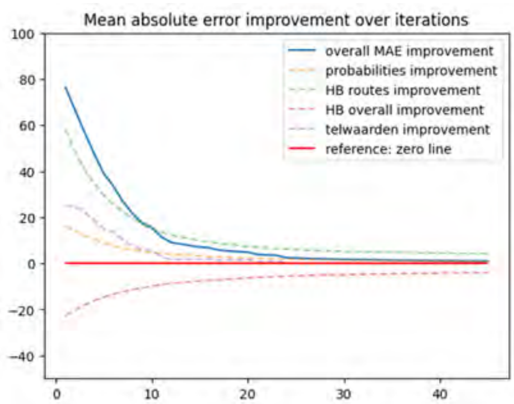

Figure 39: MAE improvement over iterations, learning rate 0.1 (repetition of figure 14)

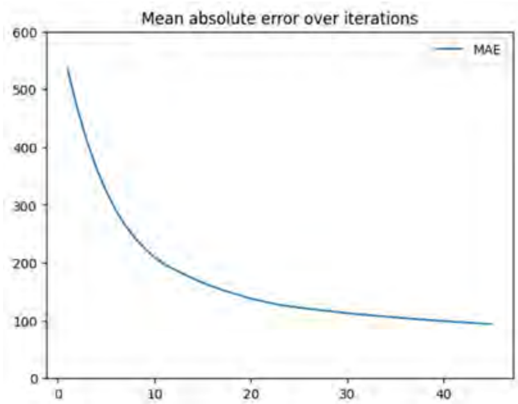

Figure 40: MAE over iterations, learning rate 0.1

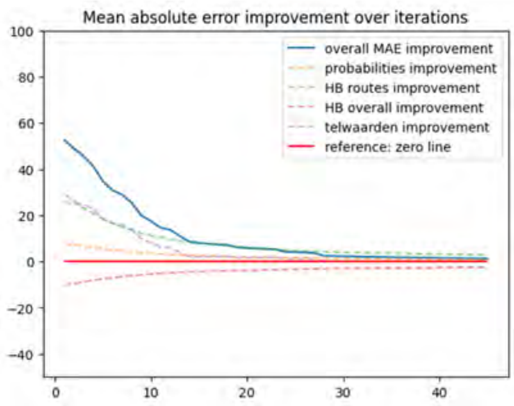

Figure 41: MAE improvement over iterations, learning rate 0.05 (repetition of figure 15)

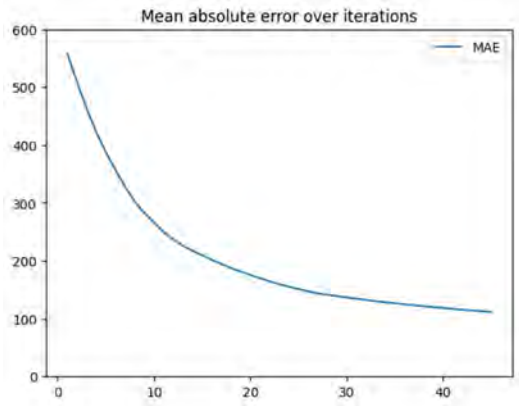

Figure 42: MAE over iterations, learning rate 0.05

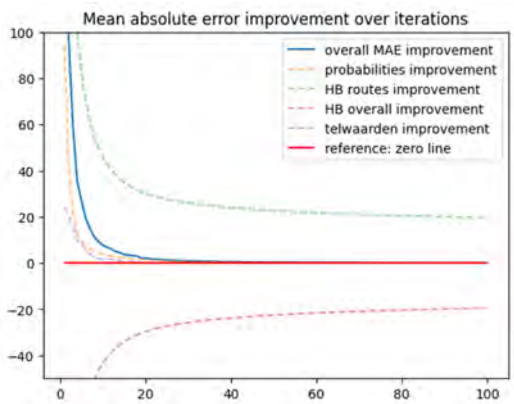

Figure 43: MAE improvement over iterations, learning rate 0.5, 100 iterations

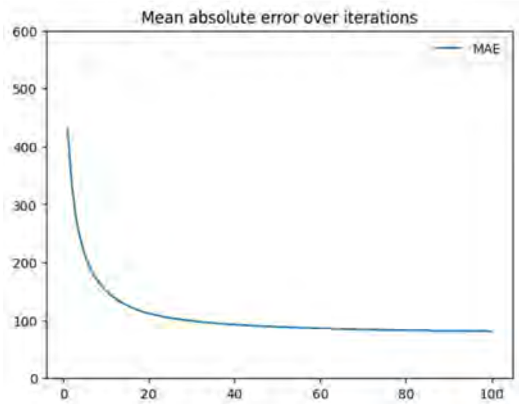

Figure 44: MAE over iterations, learning rate 0.5, 100 iterations

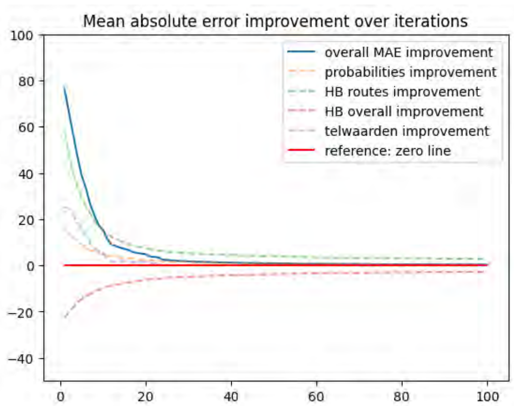

Figure 45: MAE improvement over iterations, learning rate 0.1, 100 iterations

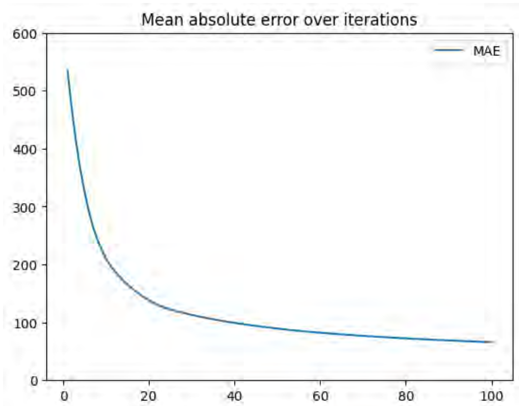

Figure 46: MAE over iterations, learning rate 0.1, 100 iterations

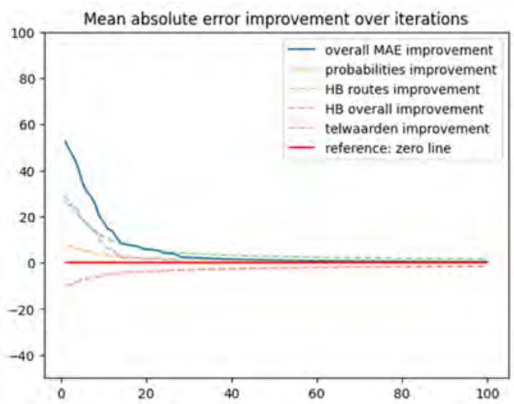

Figure 47: MAE improvement over iterations, learning rate 0.05, 100 iterations

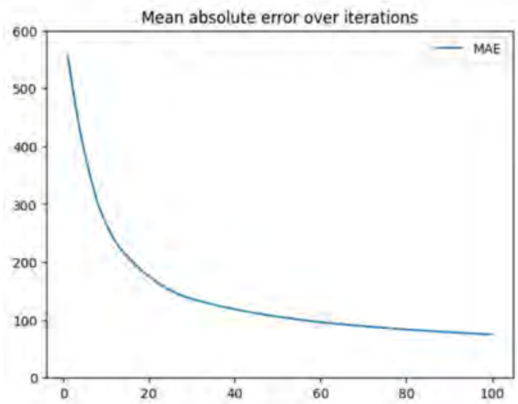

Figure 48: MAE over iterations, learning rate 0.05, 100 iterations

B Appendix B: sensitivity analysis

B.1 Sensitivity to OD-matrix

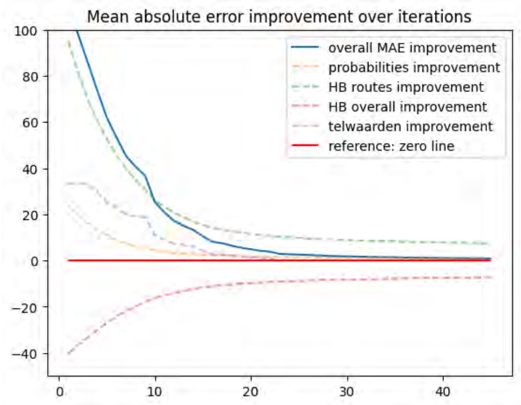

Figure 49: MAE improvement over iterations, different OD-matrix (repetition of figure 16)

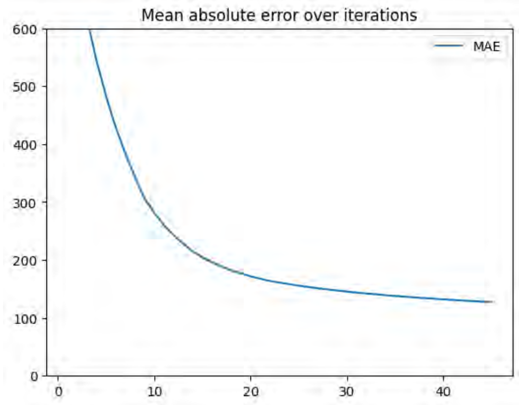

Figure 50: MAE over iterations, different OD-matrix

B.2 Sensitivity to reliability factor

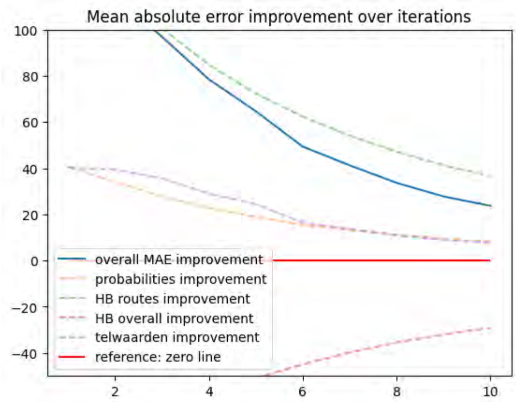

Figure 51: MAE improvement over iterations, all weight factors equal 1 (repetition of figure 18)

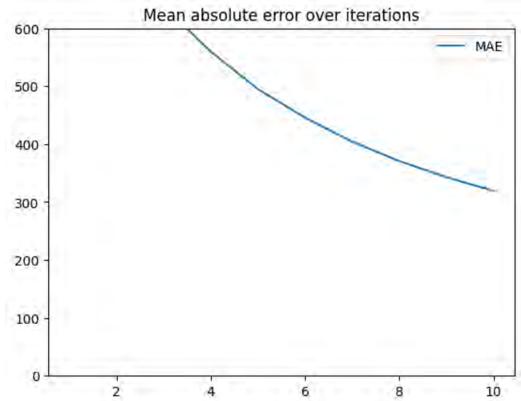

Figure 52: MAE over iterations, all weight factors equal 1

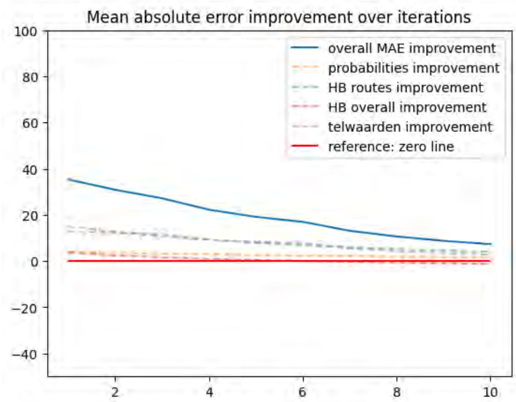

Figure 53: MAE improvement over iterations, all weight factors halved (repetition of figure 19)

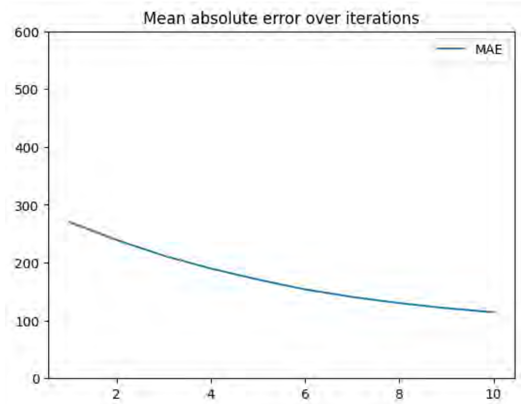

Figure 54: MAE over iterations, all weight factors halved

C Appendix C: QGIS model output

C.1 Bicycle traffic flow

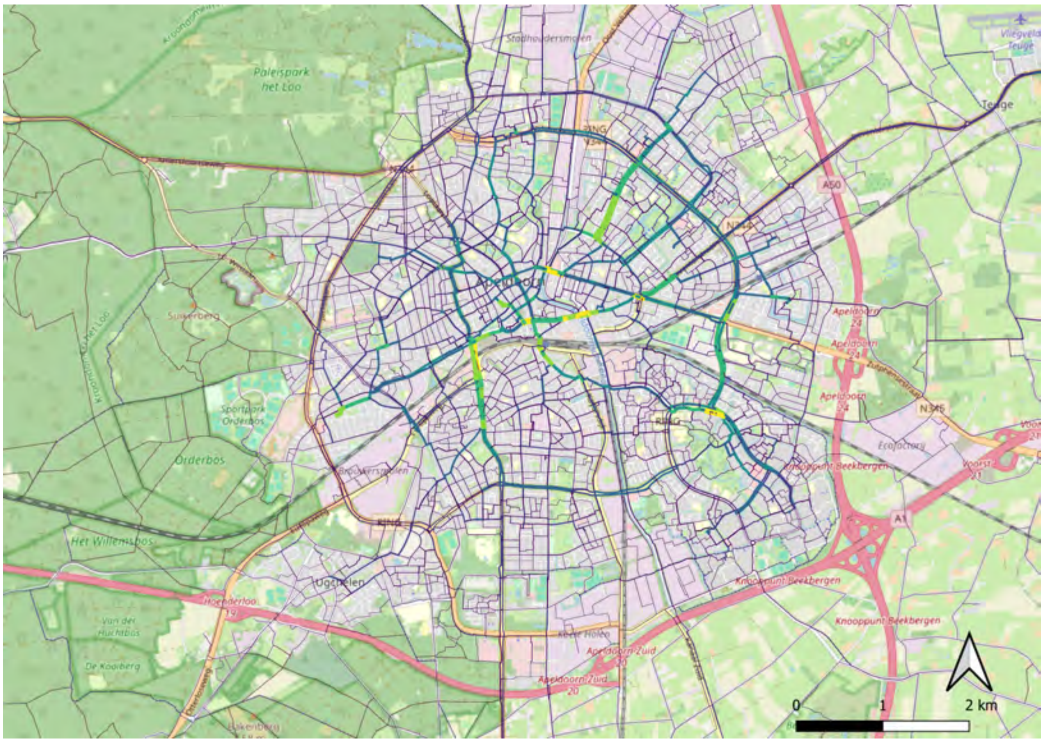

Figure 55: Uncalibrated bicycle traffic intensities

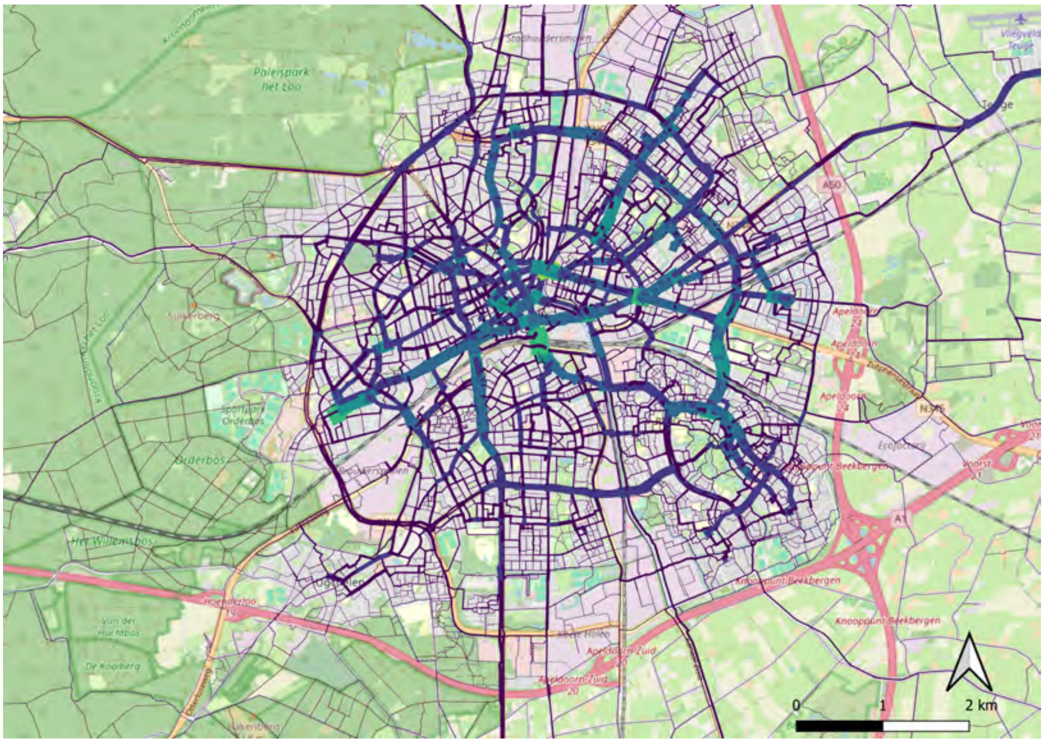

Figure 56: Calibrated bicycle traffic intensities

C.2 Count data locations

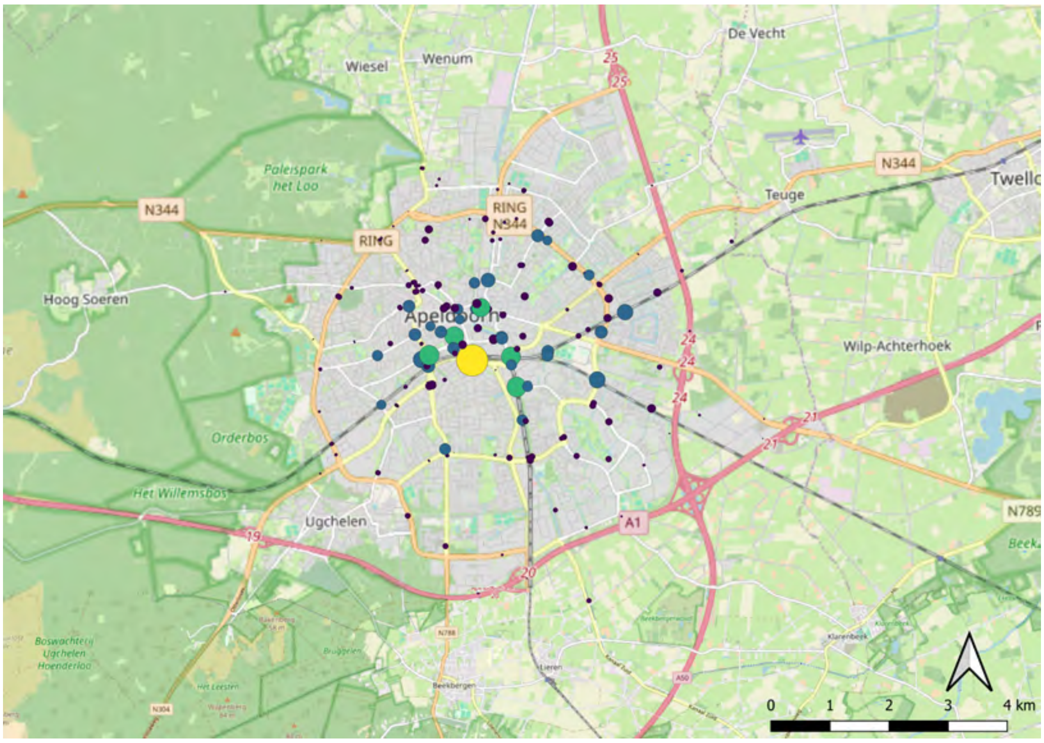

Figure 57: Uncalibrated count data

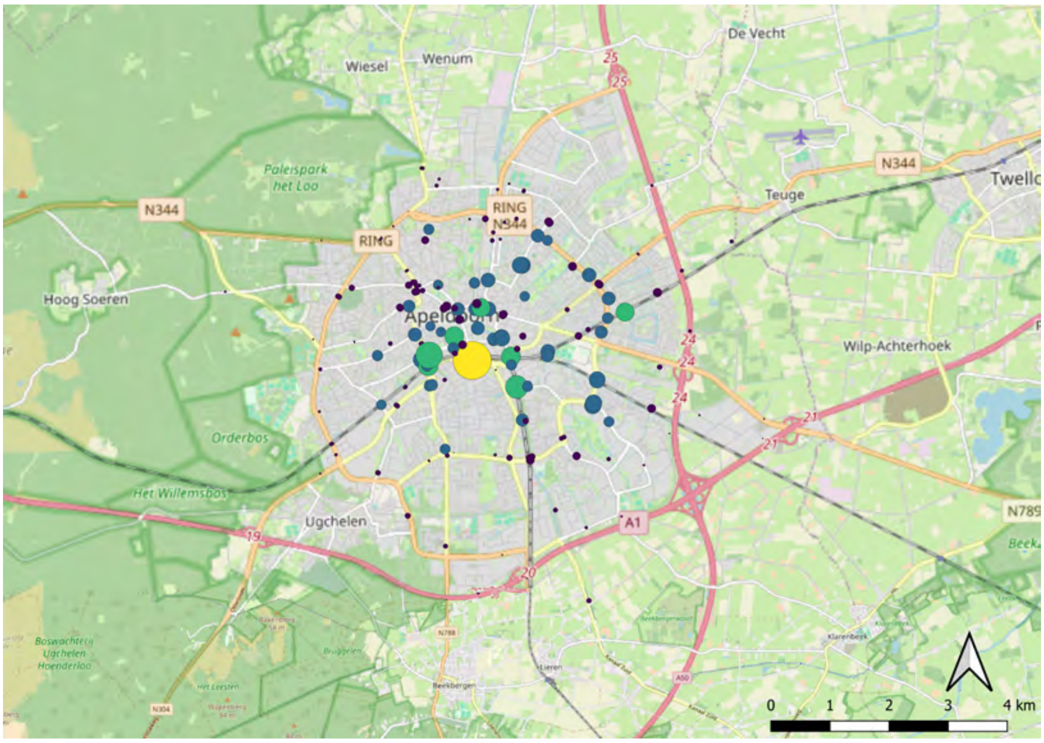

Figure 58: Calibrated count data