Estimating congestion and traffic patterns when planning road work

by

Vosse Meijssen

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	4593901	
Project duration:	April 3, 2023 – April 5, 2024	
Supervisors:	Prof. dr. ir. C. Vuik,	TU Delft
	Dr. A. Heinlein,	TU Delft
	Ir. L. Portengen	CGI
	Ir. F. de Vries	CGI

TU Delft

Contents

Nomenclature

Abbreviations

Abbreviation	Definition	Page
DTA	Dynamic Traffic Assignment	2
DUE	Dynamic User Equilibrium	3
FR	Fundamental relation	4
LWR model	The Lighthill-Whitham-Richards traffic model	1, 3
ML	Machine Learning	1
NDW	Nationaal Dataportaal Wegverkeer, the Dutch databank that handles traffic data	1
OD matrix	Origin Destination matrix	2
TA	Traffic Assignment	2
UE	User Equilibrium	3
MSE	Mean Square Error	19

Symbols

Symbol	Definition	Unit
f	The flux of vehicles through a point	[veh.$/s$]
q	The density of vehicles on a (piece of) road	[veh.$/m$]
q_c	A threshold density from which all vehicles drive at the same speed, and from which traffic flow will behave differently	[veh.$/m$]
q_j	The density at which traffic comes to a full stop	[veh.$/m$]
t	Time	$[s]$
u	The speed of vehicles on a (piece of) road	$[m/s]$
u_0	The speed of vehicles on an empty (piece of) road	$[m/s]$
x	The distance along a one-dimensional road	$[m]$

Abstract

This study investigates the impact of road alterations on traffic. Traffic congestion poses a significant challenge to transportation networks, encouraging the development of accurate traffic models to pinpoint and address these problems. Traffic flow dynamics can be researched through macroscopic traffic models such as the Lighthill-Whitham-Richards (LWR) model. Estimating the fundamental relation within the LWR model has historically been a complex matter due to the unpredictable nature of traffic. Recent progress in machine learning (ML) and big data analytics has opened new doors for improving traffic modeling accuracy.

In this study, we propose a novel approach that integrates the LWR model with an artificial neural network to estimate the parameters of Smulders' fundamental relation based on lane configurations and speed limits. Rewriting this fundamental relation allows us to use gradient descent on the parameters of that neural network. Utilizing historical traffic data from the Nationaal Dataportaal Wegverkeer (NDW), we develop a predictive traffic model that can predict the parameters of the fundamental relation.

The validity of our approach is demonstrated through the performance of the aforementioned artificial neural network, and through validation exercises. We achieve promising results, with our predictive model exhibiting an R2 score of 0.7942 in predicting traffic flux. Additionally, we implement a numerical approximation technique, the Godunov scheme, to simulate traffic flow under varying road situations. The full implementation in python can be found on `github.com/vossemeijssen/macroscopic_traffic_model`. Our findings suggest that our integrated approach offers a viable solution for predicting traffic congestion resulting from road alterations. However, the traffic model still needs a performance measure to compare it against other implementations.

While this study contributes to the body of knowledge on traffic modeling, there remain opportunities for future research, including an extension of the model to encompass network-wide traffic dynamics. An other opportunity lies in conducting deeper analyses of the impacts of alterations on travel times and congestion intensity. Still, this work represents a step forward in using machine learning techniques to improve the predictive capabilities of macroscopic traffic models.

Preface

You are currently reading the product of a year long project about traffic modeling. Looking back, I am proud of the journey that this thesis has taken. We have combined various elements from my studies - numerical approximations, neural networks, rewriting equations - all in the context of traffic.

First and foremost, I owe my gratitude to my supervisor, Alexander, from the mathematics faculty. Together we have navigated through the world of traffic theory, which was not familiar for us both. When the subject of the project became more about numerical mathematics, he was able to help me even more.

I also want to extend my thanks to the iAMLab team from CGI, especially Leonoor and Folkert. Their constant supervision and industry connections have been amazing. Furthermore, it was just great to have colleagues to work with.

To everyone involved in this project, I want to express my appreciation for your patience and flexibility. Balancing the demands of this thesis, both academically and professionally, with my commitment to rowing at an international level was no small feat. This allowed me to win a gold medal at the World University Games in Chengdu, all while putting a great end to my master applied mathematics. It was amazing, thank you all.

1 Introduction

Traffic congestion is a big problem for transportation networks. It is one of the main incentives for modelling traffic, because good traffic models can help with pinpointing problems in the network. Over the past 70 years the understanding of traffic has steadily increased and traffic models have become better in modelling traffic in a realistic way, even though traffic will always be unpredictable to a certain level. Modelling traffic requires a deep understanding of the behaviour of traffic flow in certain circumstances.

The fundamental relation has been a big focus of traffic flow analysis since the 1955 paper of Lighthill and Whitham, in [12]. Their traffic model approached traffic as a continuous flow through a network, with a positive real-valued density and speed on every point. Such a model is called a macroscopic traffic model, and the model developed in [12] is called the Lighthill-Whitham-Richards (LWR) model. The LWR model consists of a few equations, one of which is the fundamental relation of traffic. This relation has been a research topic for numerous decades [23].

Estimating the fundamental relation poses significant challenges due to the complex and random nature of traffic. Modern research subjects like machine learning and big data have helped the development of traffic model research [23]. The combination of historical data and different machine learning (ML) algorithms have shown great results in predictive traffic models [10, 1]. With the correct implementation, we can use ML algorithms to find patterns in historic data.

In this research, we propose a new approach that combines the LWR traffic flow model with ML algorithms to estimate the fundamental relation. We will build this macroscopic traffic model by using a part of an existing implementation of the LWR model. The fundamental relations are estimated using open historical traffic data, available through the web portal of Nationaal Dataportaal Wegverkeer (NDW, [25]). The end goal of this project is predicting the traffic congestion for certain road work projects. The traffic flow model should be able to model traffic in realistic work road scenarios, e.g. where one lane on a highway is closed down.

We have structured this report to systematically examine the application of machine learning in a traffic model. We will discuss the relevant theoretical background in the next chapter. Chapter 3 and 4 describe the data sources and methodology respectively. The findings will be presented in chapter 5 and discussed in chapter 6. Finally, chapter 7 will show our conclusions.

This report is the result of a master thesis project for the Master Applied Mathematics on the Delft University of Technology, in collaboration with the Infrastructure and Asset Management Lab (iAMLAB) of CGI Netherlands. It contributes to the body of knowledge dedicated to traffic modeling, specifically the prediction of traffic flow using machine learning and the LWR model.

2

Theoretical background

2.1. Basic concepts in traffic modeling

Everyone is used to a weather forecast on the evening news, where high-end weather models utilize measurements to predict the weather up to a week in the future. Why is there not such a forecast on traffic congestion? Traffic is difficult to model for a few reasons. Small perturbations in traffic speeds can have big effects, like the butterfly effect. Furthermore, even the best models can't model the stochastic nature of traffic itself, which is a result of random human influence in routing, lane-switching and accidents [2, 7, 13]. Still, traffic is definitely not fully random and traffic models have improved a lot through research over the past decades.

In 1955, M. J. Lighthill and G. W. Whitham wrote the first groundbreaking paper about kinematic waves and its application in traffic models [12]. Around the same time, P. I. Richards investigated this mathematical traffic model as well [19], and the set of equations was named the LWR model (after Lighthill, Whitham and Richards). The LWR model approaches traffic as a continuous flow using a few differential equations. This approach is also known as macroscopic traffic model. The knowledge about traffic models has steadily increased until we can now distinguish three different types of traffic models;

1. Macroscopic traffic models, where traffic is approximated as a continuous flow through a network of one-dimensional tubes. The flow is described by a system of equations.
2. Microscopic traffic models, where each vehicle is modelled independently.
3. Mesoscopic traffic models, which uses ideas from the first two types [2].

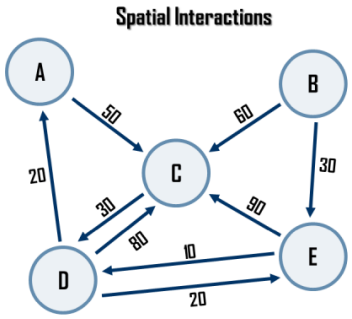
In this master thesis, we will only look at macroscopic traffic models, specifically at variations of the LWR model.

Even if these types of traffic models are very different in their approach, they often share some similarities. Usually, traffic models consist of a route choice algorithm and a simulation of traffic flow [2]. The route choice algorithm aims to solve the traffic assignment problem, defined as follows:

Definition 1 (Origin-Destination matrix) *An Origin-Destination (OD) matrix shows the number of vehicles that want to travel from one destination (represented by the row) to some other destination (represented by the column). An example can be found in figure 2.1.*

Definition 2 (Traffic Assignment) *A Traffic Assignment (TA) problem is determining how demand traffic, usually in the form of an OD matrix, is loaded onto the network. It provides a means for computing traffic flows on the network links.*

Definition 3 (Dynamic Traffic Assignment) *A Dynamic Traffic Assignment (DTA) problem is the time-dependent extension of the traffic assignment problem, able to determine the time variations in link or path flows, and capable of describing how traffic flow patterns evolve in time and space in the network [2].*



O/D Matrix

	A	B	C	D	E	Ti
A	0	0	50	0	0	50
B	0	0	60	0	30	90
C	0	0	0	30	0	30
D	20	0	80	0	20	120
E	0	0	90	10	0	100
Tj	20	0	280	40	50	390

Figure 2.1: An example of a network with 5 nodes, and a given demand in traffic between those nodes. The demand is rewritten in the OD matrix on the right. This image was taken from [26].

Related to the Nash equilibruim in game theory, John Wardrop defined a state of equilibrium for traffic models. This is known as Wardrop's principle, Wardrop's equilibrium or user equilibrium [2]. Furthermore, this equilibrium is a solution of the TA problem and also has a time-dependent version called the dynamic user equilibrium. Other solutions to TA and DTA problems are other ways that the vehicles complete their journey, but which are not in a user equilibrium.

Definition 4 (User Equilibrium) *A User Equilibrium (UE) means that the journey times on all the used routes are equal, and less than those which would be experienced by a single vehicle on an unused route. It is a solution to the TA problem.*

Definition 5 (Dynamic User Equilibrium) *A DTA problem has a Dynamic User Equilibrium (DUE) when the network has a UE at every moment. It is a solution to the DTA problem.*

In conclusion, traffic models usually consist of two components: a DTA problem and a simulation of the routes. Finding a DUE in a traffic model is a difficult task and is sometimes impossible. With macroscopic traffic models it is often possible to find this DUE as it approaches flow continuously instead of discreetly [5], but it falls out of the scope of this research. We will discuss this macroscopic traffic model further in the next section, specifically the LWR model. Afterwards, we will discuss the Godunov scheme which can approximate the LWR model numerically. This chapter is concluded with a short background of machine learning in traffic modeling.

2.2. Macroscopic traffic models

2.2.1. Lighthill-Whitham-Richards model

An example of a macroscopic traffic model is the Lighthill-Whitham-Richards model, or LWR model. This LWR model assumes a positive real-valued density of vehicles q and vehicle velocity u at every point in some network of roads. We can calculate the flux f using the definition:

$$f = qu. \tag{2.1}$$

The second equation of the LWR model is the one-dimensional continuity equation, which describes the transport of some conserved quantity. More precisely, it says that the change in vehicle density on some part of a road only depends on the in- and outflux of vehicles:

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} = 0, \tag{2.2}$$

also often notated as $q_t + f_x = 0$. This equation defines the behaviour of a conserved quantity q, where in general $q : \mathbb{R} \times [0, \infty) \to \mathbb{R}^n$ is the conserved quantity and $f : \mathbb{R}^n \to \mathbb{R}^n$ is the flux of this quantity [4]. In the case of a 1D traffic flow problem, $n = 1$. Integrating this equation on an interval $[x_1, x_2]$ gives

$$\frac{d}{dt}\int_{x_1}^{x_2} q(x,t)dx = -\int_{x_1}^{x_2} \frac{df(q(x,t))}{dx}dx = f(q(x_1,t)) - f(q(x_2,t)). \tag{2.3}$$

Equation 2.3 shows that the temporal change in amount of q inside the interval $[x_1, x_2]$ is equal to the flow entering or exiting the interval at x_1 and x_2. Because this holds for any x_1 and x_2, equation 2.2 means that the amount of q (representing the amount of traffic) can only be created at the edges of the network, and cannot be created or destroyed inside the network. This follows our intuition; traffic on a highway can only move along the highway, but (in principle) it cannot be created or destroyed there.

The third equation of the LWR model describes the velocity u as some function of the density q. This allows for a substitution into 2.2 which leads to a differential equation solely depending on the density q [2, 18]. The relation between u and q is called the fundamental relation (FR).

The idea of a fundamental relation is an important one. It relies on the assumption that the speed of traffic on some location is only dependent on the density of traffic on that location. In the real world this is of course not true. Differences in driving style and human errors can create different traffic situations, even when the conditions are identical. In traffic models, this uncertainty will always be present [7, 13]. The LWR model looks at traffic in an aggregated form and approaches it as a uniform flow, averaging over these differences.

2.2.2. Fundamental relations

The simplest example of a FR is the linear FR, where the speed of traffic scales down linearly with the density. When the density is 0, the vehicles will drive at the maximum speed u_0. When the density reaches q_j, the velocity will be 0 and traffic will come to a stop. In an equation, this reads:

$$u_{\text{linear}}(q) = u_0(1 - \frac{q}{q_j}). \tag{2.4}$$

Substituting this into 2.2 will lead to the differential equation

$$\frac{\partial q}{\partial t} + u_0 \frac{\partial (q(1 - \frac{q}{q_j}))}{\partial x} = 0$$

which, given initial conditions and boundary conditions, can be solved numerically. Of course, this linear FR is not the best representation of reality. At small densities, an increase in density will not decrease the speed that much. And at high densities, the traffic speed will not immediately drop to zero, but it will just stay very low. This has lead to a few different different FRs [2], for instance, Smulders' and De Romphs' FRs, given by:

$$u_{\text{Sm}}(q) = \begin{cases} u_0(1 - \frac{q}{q_j}), & \text{for } q < q_c \\ \gamma(\frac{1}{q} - \frac{1}{q_j}), & \text{for } q > q_c \end{cases} \tag{2.5}$$

$$u_{\text{DR}}(q) = \begin{cases} u_0(1 - \alpha q), & \text{for } q < q_c \\ \gamma(\frac{1}{q} - \frac{1}{q_j})^{\beta}, & \text{for } q > q_c \end{cases} \tag{2.6}$$

In both of these cases, the γ is chosen such that $u(q)$ is continuous. It can be seen that both of these models use the constant q_c, which is the congestion density. Densities lower than this threshold leave enough space for cars to move around eachother, which means the average speed will stay relatively constant in that density region. When the threshold has been reached, then the vehicles are stuck in a traffic jam and the average speed will drop drastically [18].

These FRs can be shown in fundamental diagrams, which are usually plots showing the relation between q and u or f. Figure 2.2 shows these three FRs in three plots, with the relations between density q, velocity u and flux f. There is not one FR applicable on every single piece of road, as the relation between density and velocity is not globally the same. For example, this relation depends on the surroundings, the behaviour of the average road user but also on the season and time of day.

It can be seen that Smulders' FR is just De Romphs' FR with $\alpha = \frac{1}{q_j}$ and $\beta = 1$, and linear FR is just Smulders' FR with $q_c = q_j$. These three fundamental relations are given in order of degrees of freedom. These added degrees of freedom can give flexibility to the model, but also adds complexity. We will focus on Smulders' FR, as this has enough degrees of freedom to model traffic accurately [18]. However, further research is needed if we want to compare the usage of different FRs in the applied implementation.

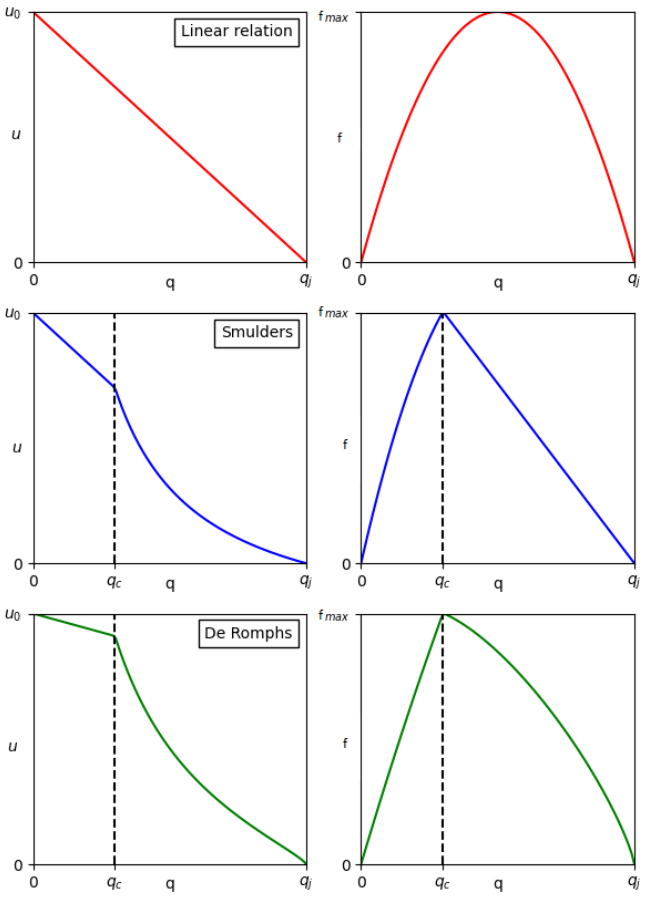


Figure 2.2: Different fundamental diagrams for three relations: the linear relation, Smulders' relation and De Romphs' relation. For simplicity in this example, $u_0 = q_j = 1$, and following [18] the other parameters have been set at a realistic value: $\alpha = 0.3$, $\beta = 0.8$, $q_c = 0.3$. In chapter 3, we can see that Smulders' FR follows reality accurately. This can also be seen in the results, chapter 5. Other FRs could have an impact on the performance of the traffic model, but this falls outside the scope of this research.

2.2.3. Rewriting Smulders' fundamental relation

Macroscopic traffic models heavily rely on the idea of a fundamental relation, or a relation $f(q)$ between the traffic density q and the flux f. It is also possible to find this relation by expressing the vehicle speed u in terms of q and rewriting it into $f(q) = u(q)q$.

Smulders' FR (given in 2.2) is dependent on three parameters: u_0, q_j and q_c. A main part of this research consists of fitting these parameters to a dataset. Finding the values of these variables for a certain piece of road is nothing more than fitting the function on historical road data, consisting of individual density-speed combinations (see chapter 3). Figure 2.3 shows the fitting of a FR based on historical data.

The fitting process is possible using an evolutionary algorithm, but convergence is slow and not guaranteed. A better fitting method would use gradient descent to optimize the three FR parameters. An easy way to use gradient descent is to see Smulders' FR as a artificial neural network, with 3 parameters, an input (density) and an output (velocity). This allows us to directly learn the parameters from historic data. The layout of the neural network and its implementation in the predictive model will be further explained in chapter 4.

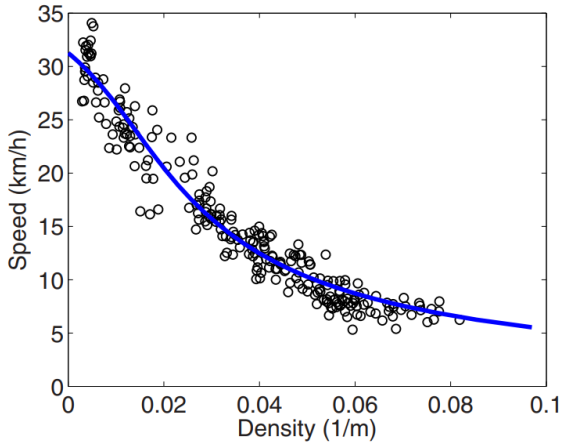


Figure 2.3: An example of a FR, fitted onto historical data. This image was taken from [8], page 249, and the FR was here derived from a model of traffic flows at intersections.

There is some difficulty fitting this speed-density function using gradient descent, coming from the variable congestion density q_c. q_c is the critical density that defines the border between two different traffic behaviours, which means the derivative $\frac{\partial \text{Cost}}{\partial q_c}$ is not immediately computable by backpropogation. We can conclude that we can't directly use Smulders' FR (eq 2.5) in gradient descent.

To make the rewriting easier, we will first remove the $1/q$ from our problem. That is why we will not fit the speed-density function, but the flux-density function. We know from the definition of flux that $f = qu$. Furthermore, continuity in $q = q_c$ means that $\gamma = u_0 q_c$. This gives the following equation for the flux, according to Smulders' FR:

$$f_{\text{Sm}}(q) = \begin{cases} u_0 q - u_0 \frac{q^2}{q_j} & \text{for } q < q_c, \\ u_0 q_c - \frac{u_0 q_c q}{q_j} & \text{for } q \geq q_c. \end{cases}$$

We will try to rewrite this into one equation using the ReLU function.

$$ReLU(x) = \begin{cases} 0 & x < 0, \\ x & x \geq 0. \end{cases}$$

Now we will rewrite both parts of $f_{\text{Sm}}(q)$. The full elaboration can be found in appendix A. Eventually, we get these expressions:

$$u_0 q - \frac{u_0 q^2}{q_j} = u_0 q_c - \frac{u_0 q_c^2}{q_j} + (-u_0 + \frac{u_0 q_c}{q_j} + \frac{u_0 q}{q_j}) ReLU(q_c - q) \text{ for } q \leq q_c$$
$$u_0 q_c - \frac{u_0 q_c q}{q_j} = u_0 q_c - \frac{u_0 q_c^2}{q_j} - \frac{u_0 q_c}{q_j} ReLU(q - q_c) \text{ for } q \geq q_c$$

We can see that both parts of the flux-density relation of the Smulders FR consist of $u_0 q_c - \frac{u_0 q_c^2}{q_j}$ and two ReLU terms which are zero in the opposite domain. This means that we can write the flux-density relation of the Smulders FR in one equation as:

$$f_{\text{Sm}}(q) = \frac{u_0}{q_j}(q_c q_j - q_c^2 + (q_c + q - q_j) ReLU(q_c - q) - q_c ReLU(q - q_c)) \tag{2.7}$$

Using this equation, it becomes a lot easier to fit the parameters. We can use a machine learning framework like Pytorch to define $f_{\text{Sm}}(q)$ as a neural network with the density of traffic q as an input, and the flux of traffic f as an output. Then, it can use backpropogation and gradient descent to optimize the parameters q_j, q_c and u_0. We will use this rewritten FR in the predictive part of our model.

2.3. Numerical solutions of traffic flow

2.3.1. Riemann problems and the Godunov scheme

As we have seen in equation (2.2), the 1D traffic flow problem is a conservation problem. Conservation problems are a type of Cauchy problems where f represents the flux of q.

Definition 6 (Cauchy problem) *The problem*

$$\begin{aligned} &q_t + f(q)_x = 0, \quad x \in \mathbb{R}, \quad t > 0, \\ &q(x,0) = q_0(x), \quad x \in \mathbb{R}, \end{aligned}$$

for some function $f : \mathbb{R} \to \mathbb{R}$ is called a Cauchy problem [4]. In this context, ***Cauchy data*** *represents the initial conditions $q_0(x)$ from which a unique solution can be found.*

According to [4, 20], it is possible to approximate the solution to this problem using a finite-difference method. For example, set $f(q)$ as the linear fundamental relation (2.4): $f(q) = q(1-q)$. This gives the following problem:

$$q_t + (q - q^2)_x = 0, \quad x \in \mathbb{R}, \quad t > 0, \tag{2.8}$$

$$q(x,0) = q_0(x), \quad x \in \mathbb{R}, \tag{2.9}$$

We will use the following discretization:

$$\begin{aligned} &x_i = ih \qquad i \in \mathbb{Z}, h > 0 \\ &t_n = nk \qquad n \in \mathbb{N}_0, k > 0 \\ &q(x_i, t_n) = q_i^n \end{aligned}$$

Using this discretization, [4] uses a few finite difference schemes to approximate the solutions of a few benchmark problems. A central, upwind, and Lax-Friedrichs scheme are applied, and the results are compared against the exact solution of these benchmark problems. The central scheme and upwind scheme both deviate from the exact solution around the shock. The Lax-Friedrichs scheme stays closest to the exact solution, using the numerical scheme:

$$q_i^{n+1} = \frac{1}{2}(q_{i+1}^n + q_{i-1}^n) - \frac{k}{2h}(f(q_{i+1}^n) - f(q_{i-1}^n)) \tag{2.10}$$

$$= \frac{1}{2}(q_{i+1}^n + q_{i-1}^n) - \frac{k}{2h}((q_{i+1}^n - (q_{i+1}^n)^2) - (q_{i-1}^n - (q_{i-1}^n)^2)) \tag{2.11}$$

There are some problems with a finite difference approximation of conservation problems [20]. This discretization method has difficulties around shockwaves, where the shock will diffuse over time or where the shock speed is calculated wrong [4]. Furthermore, some finite difference schemes like the upwind and central scheme don't implicitly conserve q. A better approach would be to use a finite volume approach like the Godunov method, introduced by Sergei Godunov in 1959 [6].

In finite volume, the area is devided into "volumes" with interfaces between them [17]. For 1-dimensional problems this means that the domain is devided into segments x_i. The Godunov method means keeping track of the amount of "conserved quantity" in each segment, and finding the flux at the segment boundaries every time step. The difficulty lies in finding this flux at every cell boundary. Setting the conserved value q_i constant in each cell gives a shock at every cell boundary. Each shock can be seen as a Riemann problem:

Definition 7 (Riemann problem) *A Cauchy problem with initial values*

$$q_0(x) = \begin{cases} q_l & \textit{for } x < 0 \\ q_r & \textit{for } x \geq 0 \end{cases} \tag{2.12}$$

where $q_l, q_r \in \mathbb{R}$ is called a Riemann problem. [9]

The discontinuity at $x = 0$ is the main focus of research about Riemann problems. In [4] this problem is solved by using the method of characteristics. They come to the conclusion that, if $q(x, t)$ is a solution, then $q(\alpha x, \alpha t)$ for some $\alpha > 0$ is a solution as well. By expressing $q(x, t) = q(\xi)$, where $\xi = \frac{x}{t}$, a few cases can be distinguished. In the light of the Godunov method, only $f(q(\xi = 0))$ is needed.

1. $q_l = q_r$ gives the constant solution $q(x, t) = q_0(x)$ and $f(q(0)) = 0$.

2. $q_l < q_r$ means that there is a higher density of traffic on the right than on the left. This higher density on the right leads to lower speeds. As traffic moves from left to right, it follows that the shockwave will stay a discontinuity. It is concluded that the solution has the form:

$$q(x,t) = \begin{cases} q_l & \text{for } x < st \\ q_r & \text{for } x \geq st \end{cases} \tag{2.13}$$

   where the shock speed s is found to be

$$s = \frac{f(q_l) - f(q_r)}{q_l - q_r}. \tag{2.14}$$

   This choice for s is called the Rankine-Hugoniot condition [4]. This means the value of $f(q(0))$ depends on the value of s; if $s > 0$, then the shock moves to the right and $q(0) = q_l$, while $s < 0$ yields $q(0) = q_r$. $s = 0$ is impossible in this situation, as that implies $f(q_l) = f(q_r)$ which is only possible if $q_l = q_r$.

3. $q_l > q_r$ has multiple weak solutions, but only one physically meaningful solution; the rarefraction wave. This means the shock will not stay a discontinuity, but it will spread out. This type of rarefraction wave is the correct solution in this situation as it satisfies the entropy condition as defined in [4] and [9]. Mathematically, this looks like this:

$$q(x,t) = \begin{cases} q_l & \text{for } x < f'(q_l)t \\ (f')^{-1}(\frac{x}{t}) & \text{for } f'(q_l)t \leq x \leq f'(q_r)t \\ q_r & \text{for } x > f'(q_r)t \end{cases} \tag{2.15}$$

   For the Godunov method, we will need $f(q(0))$. This can be found from this equation:

$$q(0) = \begin{cases} q_l & \text{if } f'(q_l) > 0 \\ (f')^{-1}(0) & \text{if } f'(q_l) \leq 0 \leq f'(q_r) < 0 \\ q_r & \text{if } f'(q_r) < 0 \end{cases} \tag{2.16}$$

   In the case of traffic models, $f(q)$ is a concave function and $(f')^{-1}(0)$ is the unique solution to $f'(q) = 0$ which represents the point of maximum flux. [4, 9]

As we told before, in finite volume the domain is devided into sections x_i with constant density q_i. On the interfaces between these sections is a Riemann problem. The Godunov scheme provides a method to calculate the behaviour of these shockwaves. The size of the time-step k should be chosen small enough such that the shock waves don't interact with eachother within the time interval. After the time-step, the flow is calculated between the sections and the density is again set to be constant in each section. This method has proven to stay closer to the exact solution in benchmark problems than finite-difference methods [6, 4].

2.3.2. Pseudocode for the Godunov scheme

This pseudocode is the Godunov scheme for scalar conservation law problems [9].

Data: Some initial $q_0(x)$, a fundamental relation $f(q)$ with maximum flux $f(q_{max})$, a domain x and boundary values $Q_{\text{boundary point}}(t)$.
Result: An approximation of the traffic flow over time.
begin
  Discretize t as t_n size k, and x as x_i size h
  Discretize $q_0(x)$ as $q_i^0 = \frac{1}{h}\int_{x-\frac{h}{2}}^{x+\frac{h}{2}} q_0(x)dx$
  for n *in timerange* **do**
    for *all* i **do**
      We will find q_i^* at the interface between x_i and x_{i+1}
      if $f'(q_i^n) \geq 0$ *and* $f'(q_{i+1}^n) \geq 0$ **then** $q_i^* \longleftarrow q_i^n$;
      if $f'(q_i^n) < 0$ *and* $f'(q_{i+1}^n) < 0$ **then** $q_i^* \longleftarrow q_{i+1}^n$;
      if $f'(q_i^n) \geq 0$ *and* $f'(q_{i+1}^n) < 0$ **then**
        $s \longleftarrow \frac{f(q_i^n)-f(q_{i+1}^n)}{q_i^n - q_{i+1}^n}$
        if $s \geq 0$ **then** $q_i^* \longleftarrow q_i^n$;
        if $s < 0$ **then** $q_i^* \longleftarrow q_{i+1}^n$;
      if $f'(q_i^n) < 0$ *and* $f'(q_{i+1}^n) \geq 0$ **then** $q_i^* \longleftarrow q_{max}$;
    for *all interior* i **do**
      $q_i^{n+1} = q_i^n - \frac{k}{h}(f(q_i^*) - f(q_{i-1}^*))$
    for *all boundary points* i **do**
      $q_i^{n+1} = Q_i(t_{n+1})$

2.3.3. Benchmark problems

To test our own implementation of the Godunov scheme, we need some way to measure the performance against other implementations given in prior researches, and against the exact solution. In the literature, there are multiple benchmark problems that are often used to test the numerical scheme. Often, they are Riemann problems with a piecewise constant initial density distribution and one initial discontinuity at $x = 0$. For simple conservation law models, the exact solution is known. This makes it easy to compare different numerical schemes against each other and against the exact solution. In [4], the first benchmark problem is

$$q_0(x) = \begin{cases} \frac{1}{4} & \text{for } x < 0 \\ \frac{1}{2} & \text{for } x \geq 0 \end{cases}$$

where, using the linear FR (2.4), the shock will move to the right with shock speed $\frac{1}{4}$:

$$q(x,t) = \begin{cases} \frac{1}{4} & \text{for } x < \frac{1}{4}t \\ \frac{1}{2} & \text{for } x \geq \frac{1}{4}t \end{cases} \tag{2.17}$$

The other benchmark problem in [4] is one with a rarefraction wave, with initial condition

$$q_0(x) = \begin{cases} \frac{3}{4} & \text{for } x < 0 \\ \frac{1}{4} & \text{for } x \geq 0 \end{cases}$$

and for $t > 0$ and the linear FR (2.4), this gives:

$$q(x,t) = \begin{cases} \frac{1}{4} & \text{for } x < -\frac{1}{2}t \\ \frac{1}{2}(1 - \frac{x}{t}) & \text{for } -\frac{1}{2}t \leq x \leq \frac{1}{2}t \\ \frac{1}{2} & \text{for } x > \frac{1}{2}t \end{cases} \tag{2.18}$$

In [4], the Godunov scheme is compared against a few finite difference methods (central scheme, Lax-Friedrichs scheme and upwind scheme). It concludes that the Godunov scheme stays closer to the exact solution than the finite difference methods, as it can be seen in figure 2.4. We will choose the Godunov scheme as a numerical approximation for the LWR model, and we will use these benchmark problems to test our implementation of the Godunov scheme.

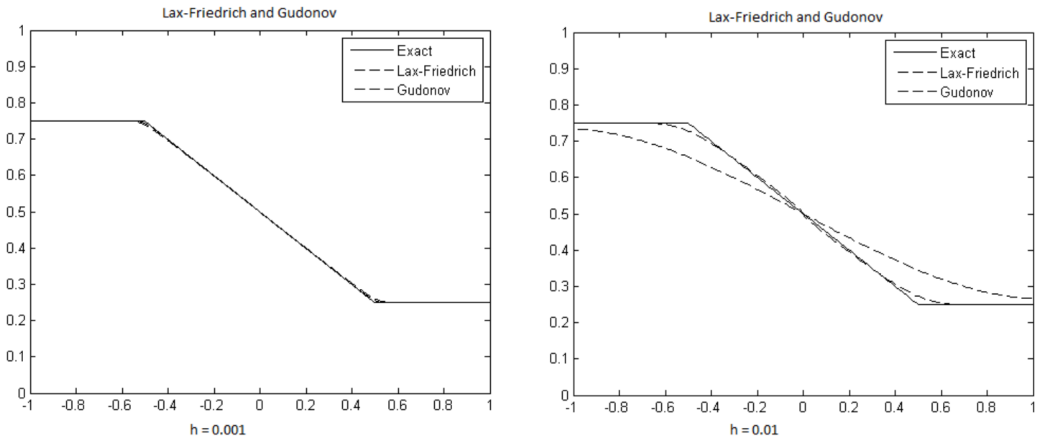


Figure 2.4: Godunov and Lax-Friedrichs scheme for a rarefraction solution with $h = 0.001$ (left) and $h = 0.01$ (right). $k = 0.001$ and $T_{end} = 1$, and the graphs depict the density (y-axis) along a one-dimensional road (x-axis). It can be seen that the Godunov scheme stays closer to the exact solution than the Lax-Friedrichs scheme. This plot is taken from [4, p. 37].

2.3.4. Testing the Godunov scheme

We implemented the Godunov scheme in Python following the pseudocode given in subsection 2.3.2. The full code can be found on `github.com/vossemeijssen/macroscopic_traffic_model`. Trying the benchmark problem given in equation 2.18, we obtain the results that can be seen in figure 2.5. This shows a rarefraction wave, identical to [4].

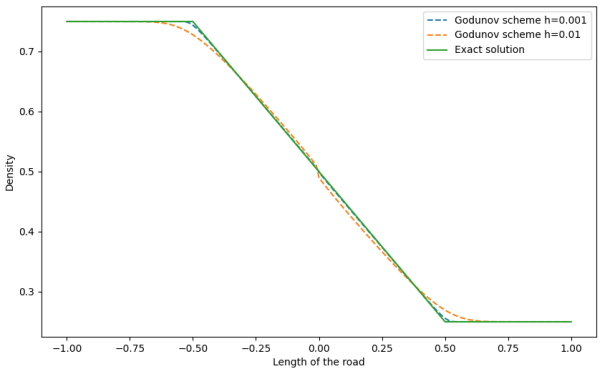


Figure 2.5: A rarefraction solution using a custom implementation of the Godunov scheme. It can be seen that the results are the same as [4], which are close to the exact solution.

The benchmark problem defined in equation 2.17 should return a moving shock wave with velocity $\frac{1}{4}$. Again, it can be seen in figure 2.6 that our implementation yields results that are comparable to [4]. The shockwave stays intact, and moves with the correct velocity.

A last test of our implementation of the Godunov scheme considers a narrowing of the road. The maximum capacity of the road is halved, which changes the fundamental relation in that part [4]. On a road with length 2, we define the fundamental relations as follows:

$$f_1(q) = q(1-q), \text{ for } x \in [0,1]$$

$$f_2(q) = q(1-2q), \text{ for } x \in (1,2]$$

Given the initial conditions $q_0(x) = 0$ and boundary conditions $q_{bl} = 0.5$ and $q_{br} = 0$, we have the necessary information to model traffic on this narrowing road. In figure 2.8 we can see the modeled density along the road. This density is similar to the modeled density in [4], which shows the implementation of the Godunov scheme is correct.

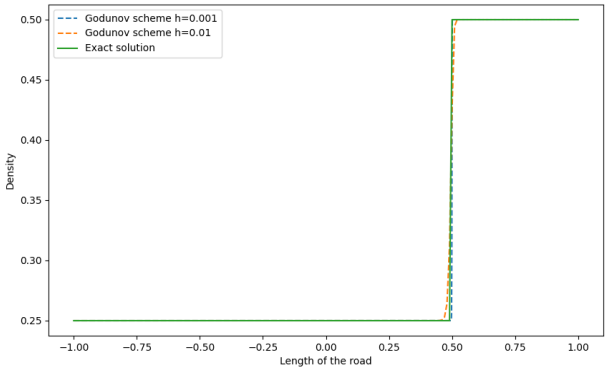


Figure 2.6: A shockwave solution using a custom implementation of the Godunov scheme. It can be seen that the results are the same as [4], which are close to the exact solution.

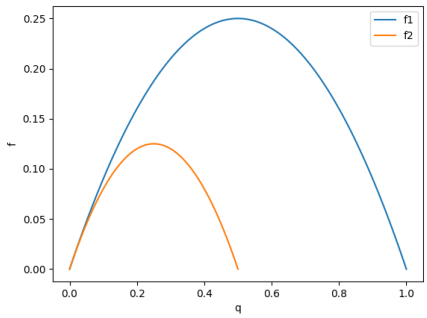


Figure 2.7: Plots of $f_1(q)$ and $f_2(q)$, that are used to model the narrowing of a road.

2.4. Machine learning

Over the past years, different Machine Learning (ML) applications have been in the center of research in different fields, and modeling traffic has been no exception. ML emcompasses a range of techniques aimed at enabling computers to learn from data and make predictions or decisions. Different ML techniques can be applied to different parts of traffic models, which are discussed in section 2.4.1.

Neural networks, a subset of ML models inspired by the structure of the human brain, have gained significant attention in recent years due to their ability to learn complex patterns in data. These networks consist of connected nodes organised in layers, with each node performing a simple computation and passing its output to nodes in the subsequent layer. Using a technique called backpropogation, we can adjust the weights of connections between nodes. That way, a neural network is able to accurately map inputs to outputs.

A common architecture for neural networks is the linear stack, also known as a fully connected neural network, where layers of nodes are arranged sequentially. Here, each node in a layer is connected tro every node in the subsequent layer, and every connection has some weight attachted to it. At each node, a combination of the weighted inputs is computed, followed by the application of an activation function. Activation functions introduce non-linearity to the network, allowing it to model complex patterns in the data. Popular activation functions include the sigmoid function, the rectified linear unit (ReLU) function, and others.

2.4.1. Machine learning in traffic modeling: an overview

Artificial neural networks form the basis for almost all predictive models in traffic modeling [24, 1]. Computer vision algorithms are used for self-driving cars or lane assisting. Other traffic forecasting models with a longer time-frame can also use machine learning algorithms to find patterns in historical traffic data, and use that to make predictions not unlike weather predictions. An example can be found on

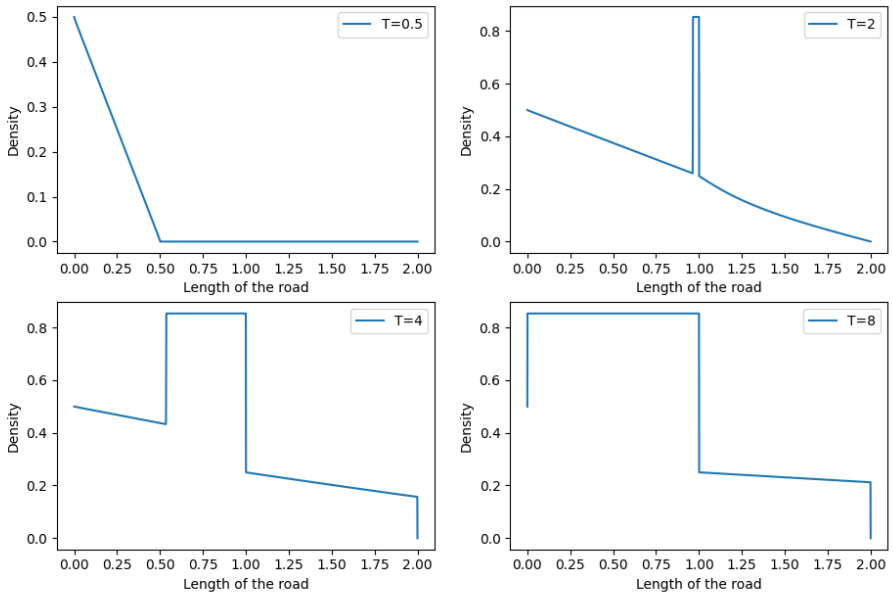


Figure 2.8: A solution modeling the narrowing of a road using a custom implementation of the Godunov scheme. These plots show four different points in time T. Traffic builds up in front of the narrowing after some time. It can be seen that the results are the same as [4].

the Keras website [10], which uses a Python implementation of an LSTM to predict the traffic situation on a single road.

We can read about the current state of deep neural networks in [21]. It recognises the ability of deep neural networks (DNNs) to predict complex patterns due to their deep structure. Furthermore, it notes that DNNs are common in research about traffic flow predictions.

Furthermore, [24] shows a big list of models that uses neural networks to directly predict traffic. Most of these models use time-dependant artificial neural networks, or DNNs for the prediction of traffic patterns. They try to predict the behaviour of traffic based on historic data.

We can also read how these forecasting models have evolved since the early 1980s. Traditionally, efforts were focussing on methods to model traffic characteristics and predict anticipated traffic conditions. They relied mostly on single-point data from highways, and were used to predict traffic volumes or traffic times. We can read that the developedment of technologies in the past years, including the availability of powerful computers and data, have enabled researchers to tackle short-term traffic forecasting in different ways.

2.4.2. Using machine learning techniques to predict the effect of road work

The way that road work affects traffic is not well understood yet. Even though road work on highways in the Netherlands is planned months in advance, it is not clear how much effect this work will have on the traffic situation. For example, we don't have a model that estimates how much total time will be lost because of these adjustments like lane closures or speed limit reductions. This research aims to fill that gap by creating a predictive model that can estimate how road alterations will affect traffic.

Existing literature has explored different aspects of traffic modeling and the application of predictive models. Looking at that existing literature, it should be possible to model alterations to the road situation. However, we have not found any study that specifically focusses on the effect of road alterations on traffic flow. There have been other studies that focus on other effects of road work, like safety [22] and the total global warming gas emissions [11]. This means there's a big opportunity for new research to develop better ways to predict how road changes will affect traffic, which could be really helpful for planning road maintenance and reducing disruptions for drivers.

3 Data

3.1. NDW open data sources

NDW is the Dutch national portal for road data. Different parties collect, combine and distribute their road data here. A big portion of this data is easily available by registering their online historic database called Dexter [25]. It is then possible to create data exports for different data sources from different domains. We will use two different NDW data sources in this project, which will be discussed in the following sections.

3.1.1. Speeds and intensities

If we want to fit a fundamental relation on some historical data using equation 2.7, we will need instances with some density q and flux f. Luckily, NDW has a data source for this type of data called "snelheden en intensiteiten", or speeds and intensities. This dataset consists of individual measurements of average speed (km/h) and average flux (vehicle/h) for a piece of road.

The speeds and intensity of traffic can be measured by different devices, like measuring loops, radar or infrared sensors [16]. These devices measure the average velocity of all cars that past over them, and that historic data can be requested through Dexter [25]. In this thesis, we only take measurements from loopdata into account. An example image of measuring loops is shown in figure 3.1.

The speeds and intensities dataset can be exported as a table with multiple columns, which can then be downloaded as a csv file. The columns that we will use in this research can be found in table 3.1. If we look at the measuring location ID, we can figure out the exact location on the highway. For example, RWS01_MONIBAS_**013**1hr**l0117**ra is the ID of a measuring loop on the Dutch highway A13, on the L direction, around hectometer 11.7. All loopdata measuring locations on the A13 use this format.

In the data, we can read the average flux and average speed in each interval. This means we can easily calculate the average density from $f = qu$. We can now plot a graph like figure 2.3 by choosing one measuring location, and fetching the average speed and average flux, and calculating the average

Figure 3.1: An example image of loops in the road, from `www.heijmans.nl/`. These loops are used to measure the speed and intensity of traffic driving over it.

Name	Explaination	Example
id_meetlocatie	ID of the specific measuring location.	RWS01_MONIBAS_0131hrl0117ra
start_meetperiode	Timestamp of the start of the datapoint.	2023-10-16 00:00:00
eind_meetperiode	Timestamp of the end of the datapoint.	2023-10-16 00:00:01
gem_intensiteit	Average flux in the interval (vehicle/h).	360.0
gem_snelheid	Average speed in the interval (km/h).	107.0
rijstrook_rijbaan	Location of the loop on the road.	lane1
voertuigcategorie	Type of vehicle. We will only use ”anyVehicle”.	anyVehicle

Table 3.1: Different columns in the NDW export ”speeds and intensities”.

density for every minute. This is shown in figure 3.2. In this figure, it can be seen that there is still a large spread in speed, especially at very low densities. We can attribute this spread to the randomness of traffic. Furthermore, the spread in velocity is bigger when the road is quiet; it is very difficult to drive 180 km/h when it is busy on the road. The spread in figure 3.2 can be reduced by filtering specific data. This filtering will be discussed in section 3.2.

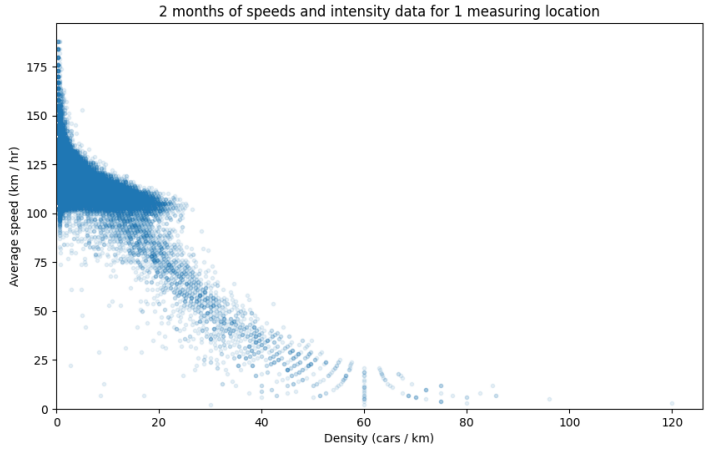


Figure 3.2: Average density and average speed on a single loop on the A13. We are looking at data from measuring location RWS01_MONIBAS_0131hrl0117ra, and only the left-most lane. Every minute of october and november 2023 is shown as a single datapoint.

We can further visualise the historical speeds and intensities by showing the densities for a piece of road. This can be seen in figure 3.3. In this figure, we see the end of the morning rush hour at the bottom, and some regular, free-flowing traffic at the top. It can be noted that the congestion moves back along the highway, which is shown by yellow diagonal lines at the bottom. This is a known phenomenon of traffic. Furthermore, we can see from the diagonal lines at the top that free-flowing traffic drives the distance between Delft and Rotterdam in approximately 8 minute.

3.1.2. Matrix sign information

In this research, we are looking for the effect of alterations to the road situation, i.e. change in maximum allowed speed or closing down lanes. If we want to learn this effect from data, then we will first need to know historical road situations. There is another NDW data source that we can use to find individual road situations, and that is the MSI data. This is dataset regarding the historic layout of matrix signs above highways in the Netherlands [16].

There are a lot of matrix signs above highways in the Netherlands, especially in the highly populated areas. These matrix signs show information like the maximum allowed speed, or whether a lane is shut down. An example of these signs can be seen in figure 3.4. The MSI data is a csv where each row depicts a change in one matrix sign. The relevant rows are shown in table 3.2. The ”Beeldstand” is the most relevant for this research, and shows the maximum allowed velocity on the lane under this matrix sign.

Because the MSI data only shows changes in the signs, it takes some steps to find the road situation

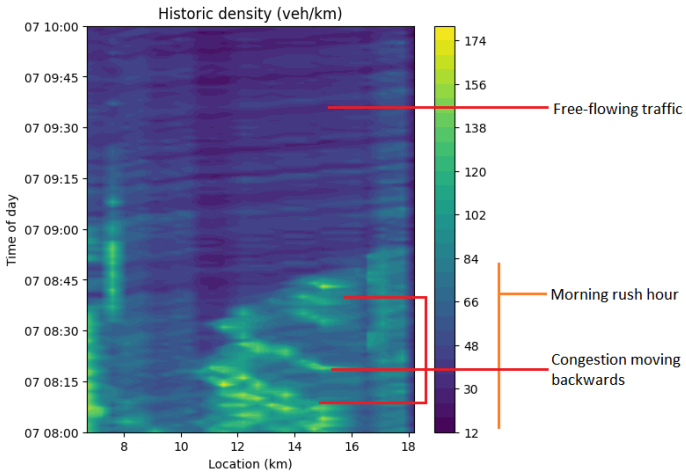


Figure 3.3: The average density in vehicle/km of traffic on the A13 between Delft (left) and Rotterdam (right), on November 7^{th} 2023 between 8:00 (bottom) and 10:00 (top). This is the end of the rush hour, and we see congestion until 8:50.

Figure 3.4: An example of a road situation, shown by matrix signs, from `www.autoblog.nl/`. In this situation, there are two open lanes with a maximum allowed speed of 70 km/h. In my model, this would reflect as $[70, 70, 0, 0, 0, 0]$.

for a specific location and time. We want to know the road situation for every instance of historical flux-density, which means we have to find a method to find the MSI data based on a location and time. Every instance of historical flux-density data has a highway number, a timestamp, and a direction. This means we can find the road situation by walking back along the highway, finding the closest matrix sign, and looking for the latest update to that specific sign. In this research, the road situation is an array of the latest updates of those signs.

This algorithm of adding MSI data to the historical speeds and intensities sounds easy, but was computationally heavy. It meant we had to look through the whole MSI dataset once for every instance of flux-density data. To optimize this, we have implemented another method that walks through the MSI data once, and fills in all speed-intensity records. This prevents walking through the MSI data an unnecessary number of times.

We want our road situation data to have the same shape for every instance in our data set. To accomplish this, we pad the roads with ”closed” lanes on the right hand side until we have 6 lanes in the array. This means that $[100, 100, 100, 100, 0, 0]$ is a highway with 4 open lanes with a maximum allowed speed of 100 km/h, while $[0, 80, 80, 0, 0, 0]$ is a highway with 2 open lanes, 1 closed lane on the left, and a maximum allowed speed of 80 km/h.

Using the algorithms and the historical MSI data, we can now plot an example of this MSI data. In figure 3.5 we can see the historical MSI data for the A13 in the morning of November 7^{th} 2023. Comparing this to figure 3.3, we see that the matrix signs react to traffic. When the density increases and it becomes busy on the road, then the Dutch traffic controllers will often lower the maximum allowed speed.

3.2. Data quality and filtering

We are able to use NDW data to combine historical records of speeds and intensities with MSI data. Eventually the goal is to predict the behaviour of traffic (which are the parameters of the FR) based on

Name	Explaination	Example
Datum en tijd beeldstandwijziging	Timestamp of the change in sign.	2023-10-15 18:50:21
Wegnummer	The highway number under the sign.	A13
DVK	The direction of the road (R or L)	R
Rijstrook	The lanenumber under the sign.	1
Hectometrering	Hectometer location of the sign.	13.7
Beeldstand	The new figure on the matrix sign.	speedlimit 80

Table 3.2: Different columns in the NDW export "MSI".

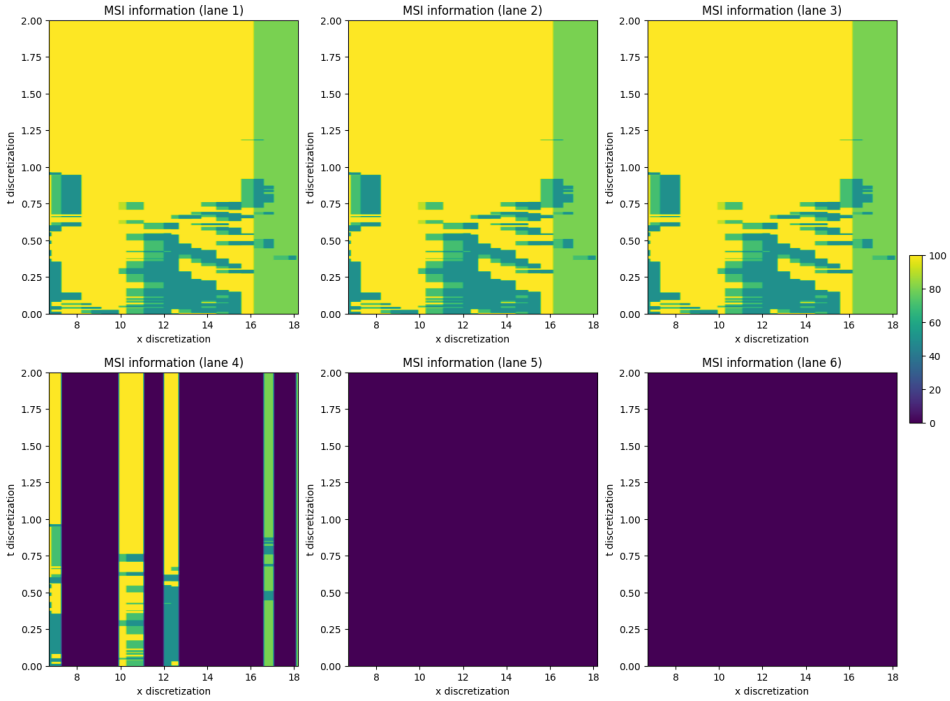


Figure 3.5: Historical MSI data from the A13. Every subplot shows the maximum allowed vehicle speed for lanes 1 through 6. This dataset consists of the A13 from Delft (left) to Rotterdam (right), on November 7^{th} 2023 between 8:00 (bottom) to 10:00 (top). Lane 5 and 6 are always 0, because this part of the highway is 4 lanes at the entries and exits, and 3 lanes in the remainder of the highway.

this MSI data. But is this a reasonable assumption?

The behaviour of traffic, aside from its randomness, can depend on multiple factors like number of lanes, maximum allowed speed, weather conditions, road condition, visibility, type of vehicles on the road, and more. In this thesis, we focus on the effect of the first two factors and try to minimize the variety in the other factors. We can minimize that variety by adding filters.

We will train and apply our model only to data from weekdays between 7:00 and 19:00. This means the ratio between work-related traffic and private traffic will mostly stay constant, but we will still get busy and quiet hours. Furthermore, all training data is from the A13 between The Hague and Rotterdam because this part contains a lot of measuring loops and matrix signs. The training and test data is taken from October and November 2023, which were both very rainy months in the Netherlands. Lastly, the data was chosen from a straight piece of highway without any entries from or exits to another highway.

The mentioned filters can be applied when creating an export in DEXTER. The speeds and intensities export consists of instances of measured average speed and flux for every minute in the domain, and for every measuring location. The first two filters are trivial: removing the data instances where no vehicle drove over the measuring loop, and removing the data instances where some error is givenl. As mentioned before, the measuring loops can distinguish multiple vehicle classes but we will only look at the "anyVehicle" class, which is the total of all vehicles that have driven over the measuring loop that minute.

The last data modification has to do with traffic flow models. It assumes one density, speed and

flux on every point along the highway, which means these models don't differentiate between different lanes. Aggregating the lanes together means adding the fluxes together, and finding the average speed using a weighted average. Given some data instance with multiple lanes l, where each lane has some flux f_l and average velocity u_l, then we find aggregate them together with the following two equations.

$$f_{tot} = \sum_{l \in lanes} f_l$$
$$u_{tot} = \sum_{l \in lanes} u_l \frac{f_l}{f_{tot}}$$

After we have applied all these filters and combining the speeds and intensities with MSI data, we can choose historical data points based on specific MSI information. Two scatter plots are shown in figure 3.6 where all historical speeds and densities are shown from measuring loops on the A13 in October and November 2023 for two specific MSI setups. The left plot mostly shows its datapoint in free-flow with low densities, while the right plot has more datapoints in the congested, "busy" domain. We can see this difference in the distribution of points along the x-axis. This can again be explained by the fact that traffic moderators adjust the matrix signs based on congestion.

The plots from figure 3.6 form the basis of the training data. We are left with a training dataset of almost 2 million rows. Eventually, we will train a model that can predict Smulders' FR based on the maximum allowed speeds per lane. This will reflect back to these graphs as a fit of Smulders' FR through these datapoints.

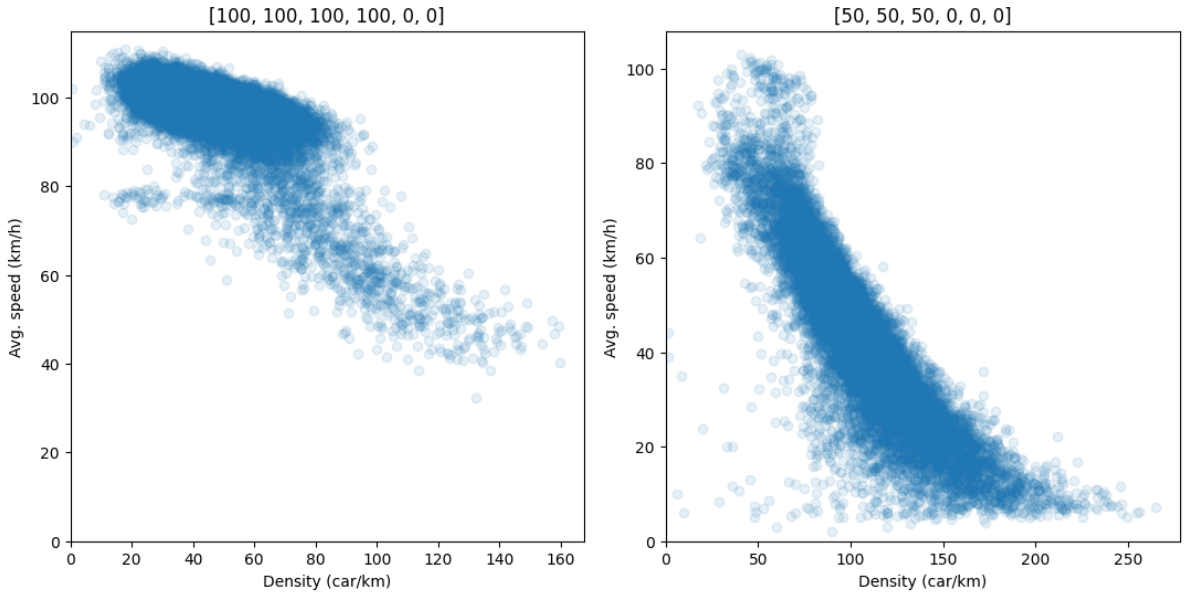


Figure 3.6: Historical speeds and intensities data, after applying all filters and aggregations, for specific road situations. The left plot shows all datapoints where there were 4 open lanes with a maximum allowed speed of 100 km/h, and the right shows 3 open lanes with a maximum allowed speed of 50 km/h.

4

Methodology

4.1. Overview of the method

The proposed method is based on a few assumptions, most of which have been discussed. It assumes traffic can be modeled with a few equations according to the LWR model. In this research, we will model traffic purely based on the available MSI information. The underlying hypothesis posits that the MSI data is a reliable predictor of the parameters of Smulders' FR.

To find the relation between MSI data and the parameters of Smulders' FR, we could use an evolutionary algorithm. However, as it was mentioned in chapter 2, this does not guarantee convergence. By rewriting Smulders' FR, we made it possible to use backpropogation and gradient descent on the parameters of the FR. This allows us to train a predictive model that finds the flux of traffic, based on the MSI data and the traffic density.

The applied method consists of three stages which can be distinguished in figure 4.1. In the first stage, it uses historical NDW data to learn the relation between MSI data and the FR parameters using an artificial neural network. In the second stage, that relation is applied to historical data with and without an alteration. The layout of these first two parts will be explained in section 4.2. Lastly, traffic is modeled in these different scenarios using the predicted FRs and the Godunov scheme, which will be discussed in section 4.3. Eventually, this method allows us to create hypothetical road situations and model them.

4.2. Layout of the neural network

4.2.1. Input and output

In figure 4.2 we have created a close-up of the learning process of the neural network. It combines historic loopdata and MSI data from NDW, and eventually predicts the parameters of Smulders' FR: u_0, q_j and q_c. The predictive part of the model lies in the artificial neural network, consisting of a linear stack of neurons with activation functions. An example of such a linear stack can be found in figure 4.3, together with an image of the softplus activation function.

In this research, we have chosen the input as 6 maximum allowed velocities and 1 density. The first instance of this input array is found in the MSI dataset for the first matrix signs back on the highway with lane number 1. The second instance has lane number 2, etc. If there is no matrix sign for that location, then the maximum speed is set as 0. All of these variables are in order of magnitude of 10^2. Furthermore, the output variable (flux) has an order of magnitude of 10^4. Neural networks tend to work better when the in- and outputs have order of magnitude 1, so in this project we scale them down by a factor of 10^2 and 10^4, respectively. These numbers are chosen through the order of magnitue in the dataset, but a more standard normalization like min-max scaling could also have been used.

Let the set of maximum allowed speeds per lanes be $l = (l_1, l_2, l_3, l_4, l_5, l_6) \in \mathbb{R}^6$, and let the set of FR parameters be $p = (u_0, q_j, q_c) \in \mathbb{R}^3$. The full predictive $F : \mathbb{R}^6 \times \mathbb{R} \to \mathbb{R}$ predicts the flux f based on l and q. First, the linear stack $L : \mathbb{R}^6 \to \mathbb{R}^3$ predicts the FR parameters p. Secondly, the Smulders' FR $f(q, p) : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}$, rewritten in equation (2.7), returns the flux for some given density. Then $F(l, q) = f(L(l), q)$. We can now look at the dataset as a matrix with $6 + 1 = 7$ input column and 1

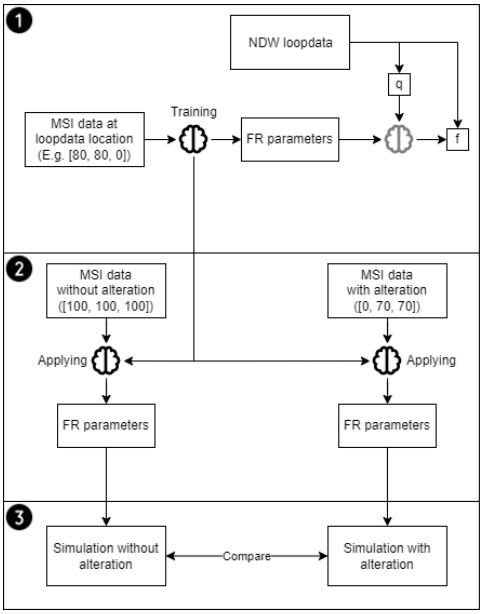


Figure 4.1: A schematic overview of the applied method. It learns the relation between MSI data and Smulders' FR parameters in the first stage. Every instance of NDW loopdata is one data point with a density q, flux f and an array of MSI data. An artificial neural network is then trained to predict the three parameters of the FR based on the MSI data. In the second stage, we apply that predictive model to find the difference between a traffic situation with and without alteration. In the final stage, we then use those fundamental relations to model traffic.

output column, because we have the density, flux, and MSI data combined for each loopdata instance.

4.2.2. Hyperparameter tuning

The used method can be modified with numerous hyperparameters. Not all of these parameters have a big impact on the learning process and the resulting predictive model. In this section, we will discuss the impact of the number of hidden layers, the learning rate and the initialization function for the weights. All hyperparameter tuning tests have been performed with a 5-fold cross validation. The Mean Square Error (MSE) loss of both the training set and test set are shown in plots. We would like to add that these hyperparameter tuning experiments are solely ranking different setups on the MSE loss of the datasets. To accurately rank different setups, we should actually look at the performance of the consequent traffic model. However, we will see in section 5.2 that this is easier said than done.

We will use Pytorch 1.13 to handle the neural network and the training data. Pytorch can be used to define basic neural networks like the linear stack with a softplus activation function, but it can also be used for backpropogation through custom functions. We need that second property for Smulders' FR, as we will define this equation as a "neural network". This means we can use Pytorch to perform gradient descent through Smulders' FR. Another feature of Pytorch is its ability to split the dataset into batches, which will have a fixed size of 1000 in these experiments. This allows the neural network to learn in small steps, rather than trying to apply gradient descent on the whole trainingset.

For the first hyperparameter tuning experiment, we have researched the effect of the layout of the neural network. Increasing the amount of hidden layers and nodes increases the capacity of the model to learn complex patterns in the data, but also increases the computational complexity of training. It also comes with a risk of overfitting. In this first experiment, we increase the amount of hidden layers and increase the amount of nodes at the same time. The parameters can be seen in table 4.1, and the MSE for different epochs is shown in figure 4.4.

It can be seen that a model with 0 hidden layers (a linear model with a softplus activation function) converges extremely fast, but is not precise. This makes sense; the neural network is just a linear model with a softm activation layer. When we add more layers, the model becomes more precise. At the same time, it converges slower and slower. When we have 4 hidden layers, the training is too slow to converge in 10 epochs. However, there is still no sign of overfitting. That makes sense, as the dataset contains almost 2 million entries.

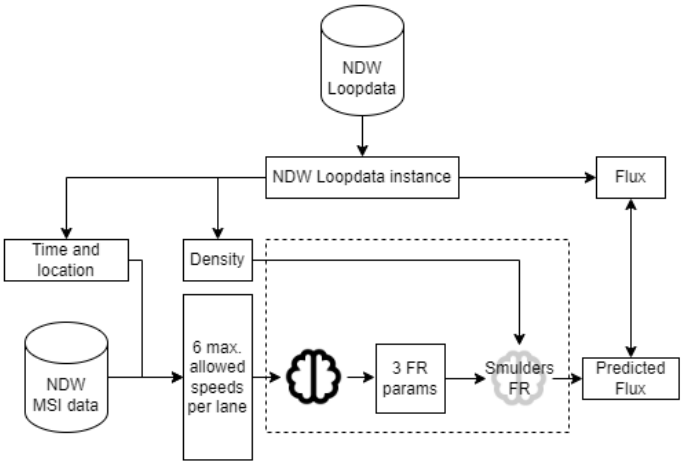


Figure 4.2: An overview of the learning process of the neural network. Every loopdata instance has a flux, density, time and location. We can find the corresponding MSI data (saved as an array of 6 floats) from those last two parameter. The array is used as in input of an artificial neural network, with the 3 parameters (u_0, q_j and q_c) of Smulders' FR as an output. We can use these parameters in combination with the density to find a predicted flux. The network will learn by finding the MSE loss between the historic flux and predicted flux, and backpropogating through the network.

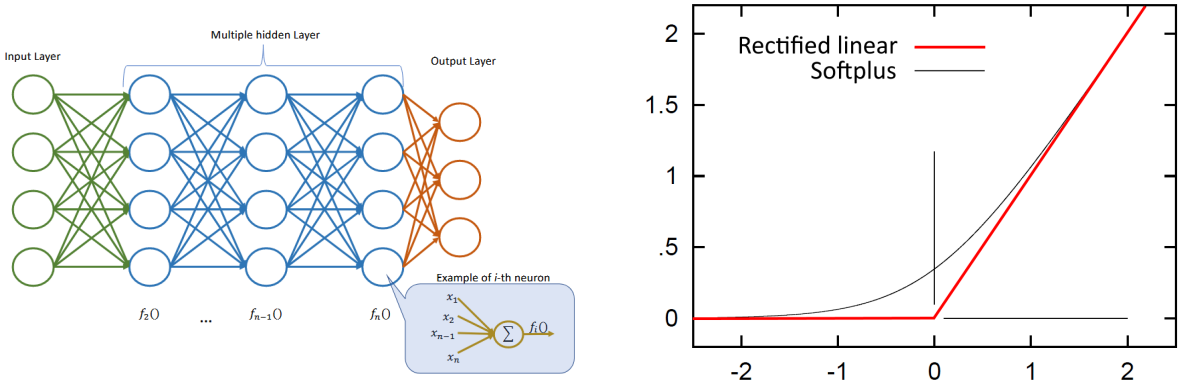


Figure 4.3: Here, we see two schematic images of parts of the predictive model. On the left is an example of a linear stack, where every layer is fully connected and where every node has some activation function [15]. On the right, the image shows the difference between a ReLU and a softplus activation function. In this research, we will use a softplus activation function in the linear stack. This second image is taken from `en.wikipedia.org/wiki/File:Rectified_linear_and_Softplus_activation_functions.png`.

In the second experiment, we will vary the learning rate. The neural network has 1 hidden layer with 4 nodes. The results of this experiment can be found in figure 4.5. In this figure, every solid line shows the MSE loss of the training set of one experiment, while every dashed line shows the MSE of the test set. Because we have 5-fold cross validation, every setup will have 5 solid and 5 dashed lines. It can be seen that the larger learning rates converge to the wrong value with a higher MSE. A learning rate of 0.05 and 0.01 converge to the best MSE we have seen of 0.0013 for all folds. The smaller learning rates have a tendency to converge to a local optimum with a MSE of 0.0018. This optimum only returns linear models, where $q_c \geq q_j$. Apparently, the network can get stuck in this local optimum when we use the wrong parameters.

To further analyse the local optimum, we have looked at the effect of different initialization functions for the weights in the neural network. We use a learning rate of 0.01, and 1 hidden layer. Pytorch has a number of built-in initialization functions. We have trained a neural network for 20 epochs using different initialization functions, the results of which can be found in figure 4.6. All functions converge to the global minimum of 0.0013, except for the zeros.

As a final hyperparameter tuning experiment, we have done a grid search where we consider all combinations of learning rates $lr \in \{0.0001, 0.001, 0.01, 0.1, 1.0\}$ and model structures with a number of hidden layers $n \in \{0, 1, 2, 3, 4\}$. The results of the grid search can be found in figure 4.7, and they are

Hidden layers	Nodes per layer	Variables
0	6, 3	21
1	6, 4, 3	43
2	6, 10, 4, 3	129
3	6, 20, 10, 4, 3	409
4	6, 40, 20, 10, 4, 3	1369

Table 4.1: The humber of hidden layers in the first hyperparameter tuning experiment.

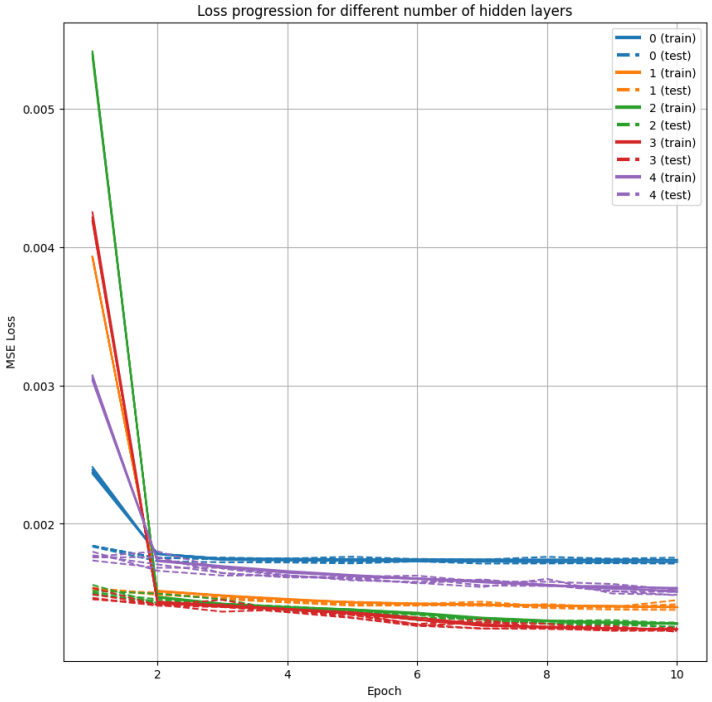


Figure 4.4: The resuls from the first hyperparameter tuning experiment, where different layouts of the neural network are trained and tested. Of course, 10 epochs is not for some training process. However, because the batch size is 1000 and the dataset contains almost 2 million instances, this means that the neural network uses gradient descent almost 10000 times in those 10 epochs. For almost all artificial neural networks in this experiment, that is enough to converge. Only the network with 4 hidden layers is still visibly improving.

similar to the other hyperparameter tuning experiments. Again, we see no sign of overfitting. Adding more layers to the neural network seems to improve the performance, but a network with 4 hidden layers learns too slow to be able to perform well after 10 epochs.

4.3. Application of the Godunov scheme

In section 2.3.4, we have implemented and tested the Godunov scheme for macroscopic traffic models, and have made sure that it can handle a single road with different FRs on different parts. The next step would be to apply the Godunov scheme to a realistic scenario. To achieve this, we discretize a highway like in figure 4.8. The boundaries x_i between the volumes are chosen evenly spaced along a highway with volume size 50 meter, and the FRs on these boundaries are found using the predictive model in combination with historical MSI data. We can use these FRs with the Godunov scheme to approximate the flow of vehicles between volumes for each time-step. In this research, we have used a time-step of 1 second. It is now possible to model traffic, given an initial density and boundary conditions.

After applying the predictive model and the Godunov scheme to historical data, we are left with a modeled version of a historic traffic situation. In figure 4.9 we can see a comparison between the historic density and the modeled density. We see the morning rush hour on the A13 from Delft to Rotterdam, between 8:00 and 10:00, November 7th. It can be seen that the model gravely overestimates the traffic. Only after 9:15, the modeled density seems to resemble the historic density. We can also see

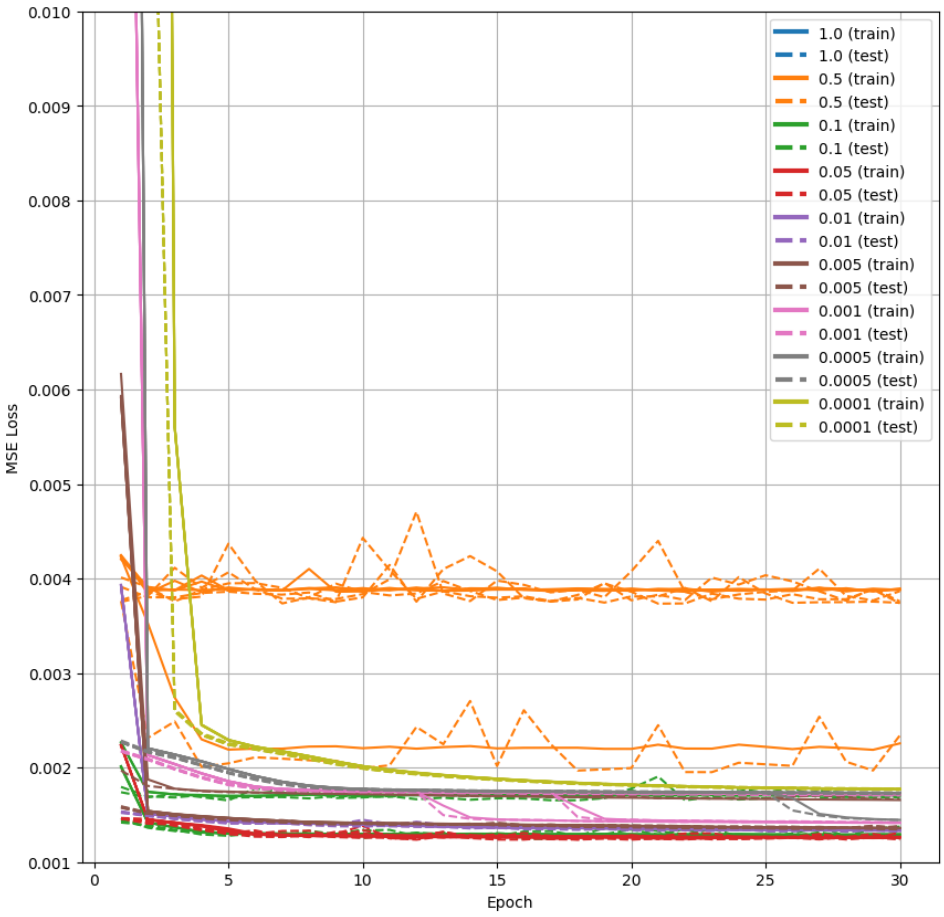


Figure 4.5: The resutls from the second hyperparameter tuning experiment, where we vary the learning rate from 10^0 to 10^{-4}.

the congestion moving backwards along the highway in both the historic and modeled density plots. All in all, this first traffic model is not really a realistic representation of reality.

There are however problems with this model; the full length of the A13 highway has some entries and exits, and our traffic model is not able to model them. When we just look at the highway from hectometer pole 12.2 up until 16.5, we have a piece of highway without entries or exits. Figure 4.10 shows the historic density and the modeled density in this domain.

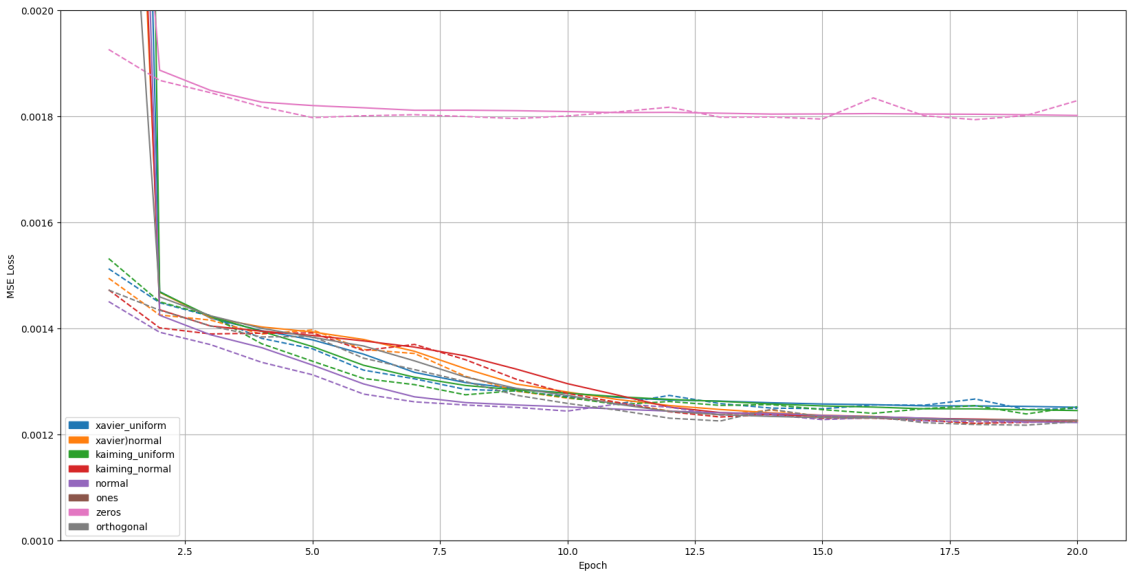


Figure 4.6: The results from the third hyperparameter tuning experiment, where we use different initialization functions to see which ones converge to a local optimum. All initialization functions are built-in pytorch functions.

Figure 4.7: The results from the last hyperparameter tuning experiment, which was a grid search. We have trained models with 5 different learning rates and 5 different model layouts for 10 epochs. The results are similar to previous hyperparameter tuning experiments, where a learning rate of 0.01 is optimal and without a sign of overfitting.

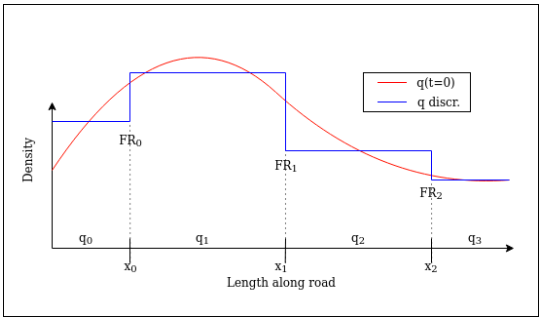


Figure 4.8: An overview of the application of the Godunov scheme. It is a finite volume method, which means each volume has a constant q_i. We can find these densities from the initial density (the red line). Every boundary x_i between volumes depicts a shock, which behaves according to some fundamental relation FR_i. Those fundamental relations follow from the MSI data and the predictive model.

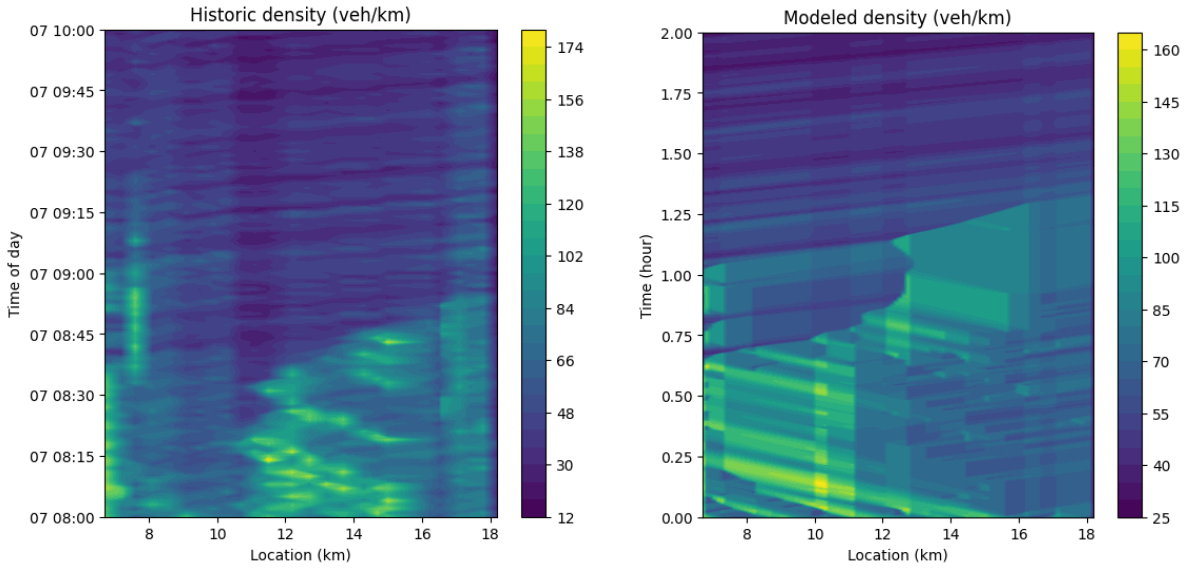


Figure 4.9: Historic density and modeled density of traffic on the A13 from Delft to Rotterdam on November 7th between 8:00 and 10:00. The modeled density overestimates the traffic congestion because of the entries and exits in this domain.

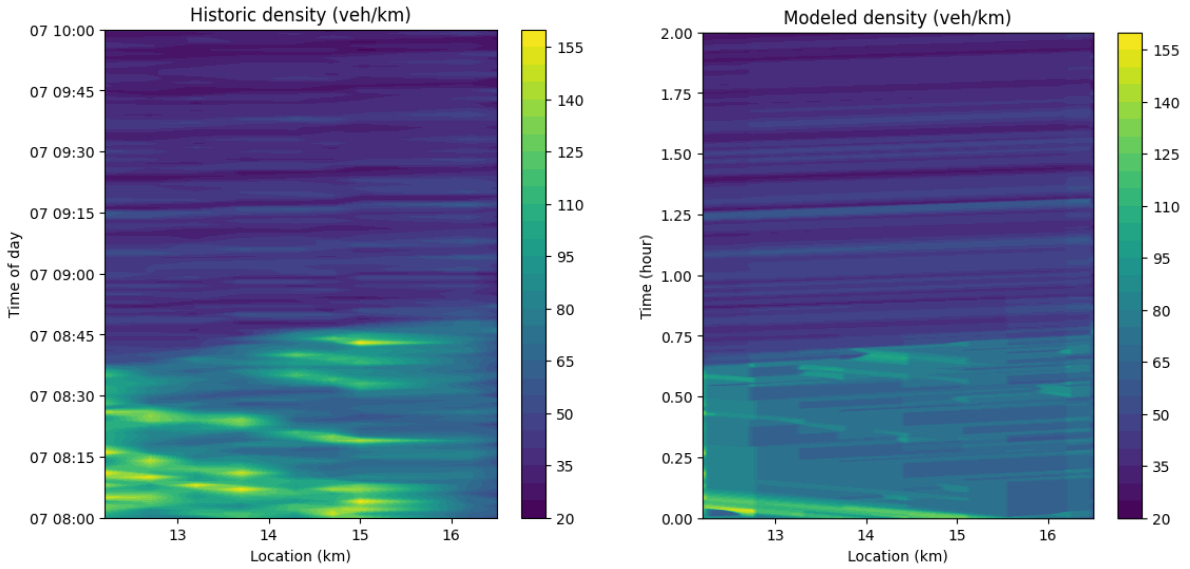


Figure 4.10: Historic density and modeled density of traffic on the A13 from Delft to Rotterdam on November 7th between 8:00 and 10:00. This time, we have only modeled the highway between hectometer poles 12.2 and 16.5, as there are no exits or entries in that part of the highway. It can be seen that the modeled density closely resembles the historic data.

5 Results

5.1. Results of the neural network

As we have discussed in chapter 4, the neural network is a predictive model with 7 inputs and 1 output. After the hyperparameter tuning, we have trained the model using 2 hidden layers, a learning rate of 0.01 and `torch.nn.init.xavier_uniform_()` as initialization function, which takes values using a Xavier uniform distribution [3]. The dataset contains all NDW loopdata from the A13, from the months October and November 2023. This dataset is split into 80% training- and 20% test-dataset.

We can now make predictions of the flux in the test-dataset using the predictive model. In figure 5.1 we can see a comparison between the predicted and historic flux. The model is able to predict the flux decently well, with an R2 score of 0.7942. A score of 0 means no correlation between the quantities, while a score of 1 means a perfect linear correlation.

We can see the predictive capabilities of the neural network when we plot the fundamental relations of different road situations. Figure 3.6 shows the historic data for two different situations. We are now able to generate a predicted Smulders' FR for each situation, which can be seen in figure 5.2. It can be seen that the fit approximates the functions quite well.

The left figure in 5.2 shows more datapoints in the freeflowing part of the FR, where $q < q_c$, while the right figure shows more datapoints in the congested part. This phenomenon can be attributed to the fact that road managers react to congestion. The maximum allowed speed will probably just be 100kmph when it is quiet on the road, but that speed will be set lower when congestion is happening.

5.2. Quality of the traffic model

In this section, we will discuss the quality of the traffic model. By this, we mean the final step in our methodology, where we try to replicate historic traffic density plots based on MSI data. Comparing how well traffic models perform against other models can be tricky because there isn't a standard way to measure that performance. There have been a lot of implementations of traffic models in previous researches [4, 9, 14, 7, 10], but they don't measure the performance of traffic models in any way, other than visually.

In our research, we are focusing on the performance of the Godunov scheme and on the performance of the predictive model. Furthermore, we will work out some example scenarios in the following section to make the results visual. It would be possible to compare different methods ourselves by choosing one or more performance metric and scoring all methods on this metric. We have chosen to leave that for further research.

5.3. Example cases of traffic prediction

We have compared historic situations against modeled situations, where we try to replicate the vehicle densities along a road based on the MSI data. Our approach to traffic modeling allows us to create different scenarios with identical initial conditions and boundary conditions, but with altered MSI data. This means we can model the closure of one or more lanes, or model the lowering of maximum allowed speed.

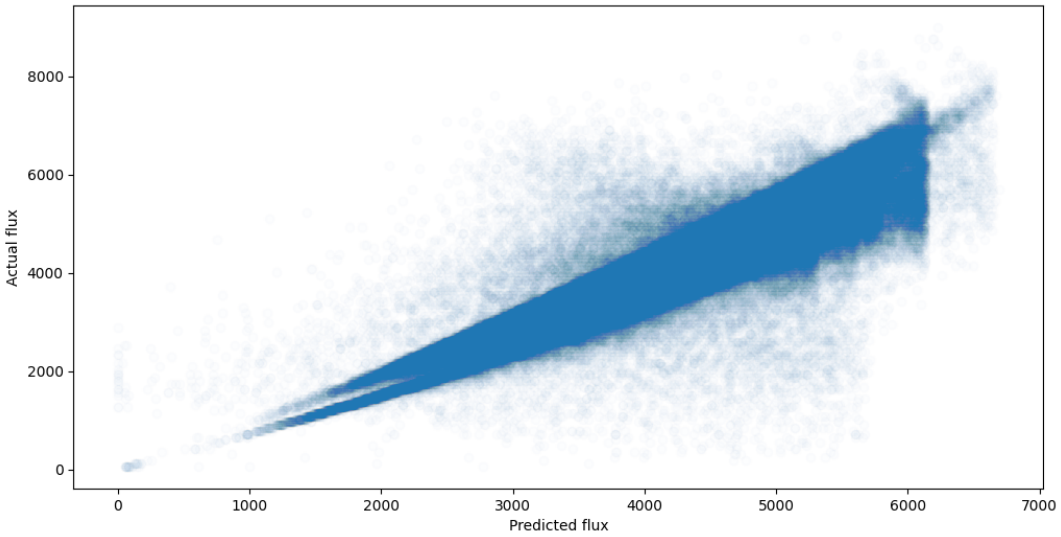


Figure 5.1: A scatter plot of the predicted flux against the actual flux in the test-dataset. The R2 score of this scatter plot is 0.7942, which means means there is a strong positive correlation between the two quantities. There is still a big uncertainty present, and we can see that this uncertainty is proportionate with the flux.

Of course not all situations are realistic or interesting. In this section, two example situations are modeled and compared to historic traffic flows. These situations are relevant for traffic management, and the result of our predictive model could help with rules and regulations regarding these situations.

5.3.1. Full stop

The first situation we will discuss, is a full stop on the highway. We will look at the A13 between 9:00 and 11:00, on November 9th. This historic data depicts a normal thursday morning, with the last part of the morning rush hour and almost two hours of free-flowing traffic. In our model, the highway will be totally closed down between 8:50 and 9:00 around hectometer pole 15.5.

In figure 5.3, we can see a comparison between the historic density, the modeled density and the modeled density with alteration. It can be seen that the modeled density closely resembles the historic density. The end of the morning rush hour is almost identical, with a backwards moving congestion at the start. After 9:15, the velocity of the free-flowing traffic is also similar. This shows that the model is able to replicate the historic situation decently well.

Adding a full stop to the model adds a big queue of more then a kilometer in front of the obstruction. After the obstruction is removed, the queue starts driving slowly and after another 10 minutes, we don't see any after-effect of the obstruction anymore in this domain. To model the complete effect of a full stop, then we should first be able to model the roads in a connecting network.

5.3.2. Closing 1 or 2 lanes

In this next subsection, we will discuss a common issue in traffic management. Every piece of highway in the Netherlands has to follow some standards when closing lanes. These are norms about the duration, the time of day, and number of closed lanes that are allowed. We can put these norms to the test by looking at the difference between closing 1 lane, and closing 2 lanes.

In this experiment, we will look at traffic from Delft to Rotterdam on the A13 on November 7th between 12:00 and 12:30. This is a normal tuesday afternoon, without any congestion or road work. In figure 5.4 we can see the historical density compared to the modeled density. The slanted line pattern shows traffic moving from Delft (left) to Rotterdam (right), and the modeled density accurately depicts the historic density and vehicle velocities. We can see that by comparing the slope of the slanted line pattern.

Now, suppose there is some road work needed between 9:07:30 and 9:22:30 on this piece of highway. We will model two hypothetical situations; two open lanes, and one open lane. We can model this by setting the MSI data for the obstructed domain to be equal to $(90, 90, 0, 0, 0, 0)$ or $(90, 0, 0, 0, 0, 0)$. The modeled densities can be found in figure 5.5. When we have two lanes available, we can vaguely see

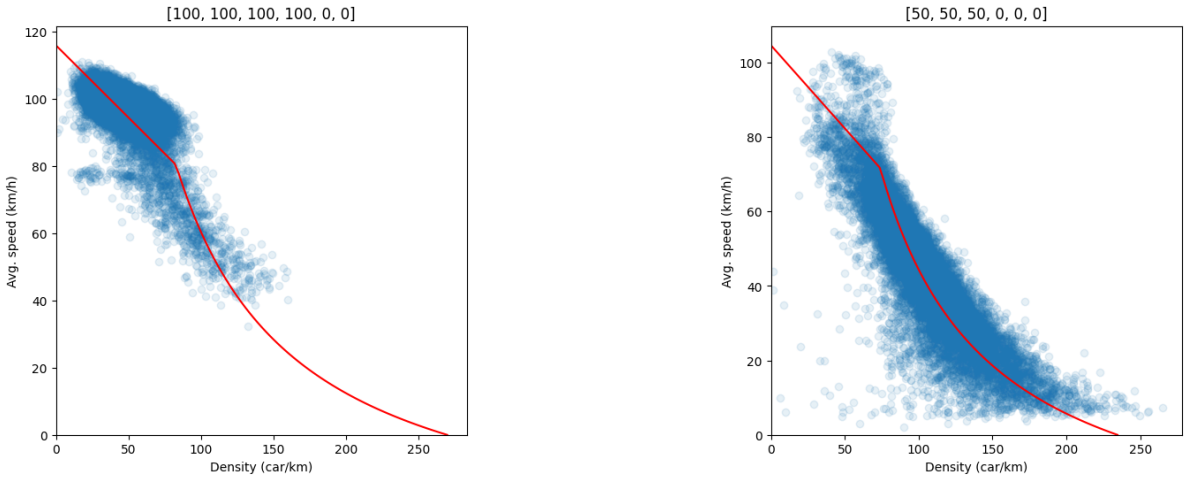


Figure 5.2: Two example subsets of the test-dataset with different MSI data; 4 lanes with a maximum allowed speed of 100kmph, and 3 lanes with a maximum allowed speed of 50kmph. The blue points are historic data of those situations, while the red lines depict Smulders' FR, as predicted by the neural network.

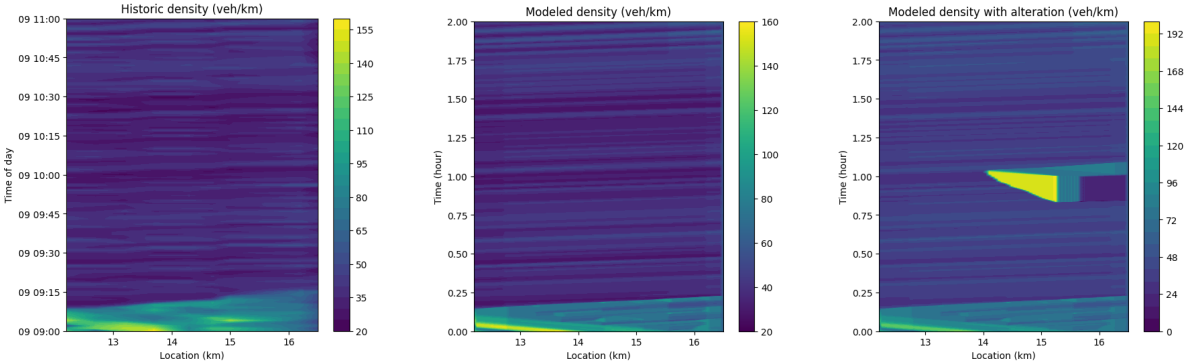


Figure 5.3: Three plots which show the density plots of the A13 between Delft Zuid and Rotterdam, on November 9th between 9:00 and 11:00. The left plot shows the historic density. We can see the end of the morning rush hour until 9:15. Afterwards, we just see free-flowing trafic which travels from Delft (left) to Rotterdam (right). The free-flowing traffic is depicted as lines with a slope of approximately 100kmph. The center plot shows the results of our traffic model, given historic MSI data and boundary- and initial conditions. The right plot depicts the modeled density in the domain, but with an alteration. We have added a full stop between 8:50 and 9:00, and we can see a traffic jam forming in front of the full stop. After the obstruction is removed, the traffic jam resolves swiftly.

the opstruction in the density plot, but traffic is still flowing freely. Closing down another lane has bigger implications on the flow of traffic. We can then clearly see an increase in density and a slower vehicle velocity along the obstructed area. Of course, these plots themselves are not enough to check rules and regulations regarding the closure of lanes, but they can be use as a starting point for calculating additional travel time.

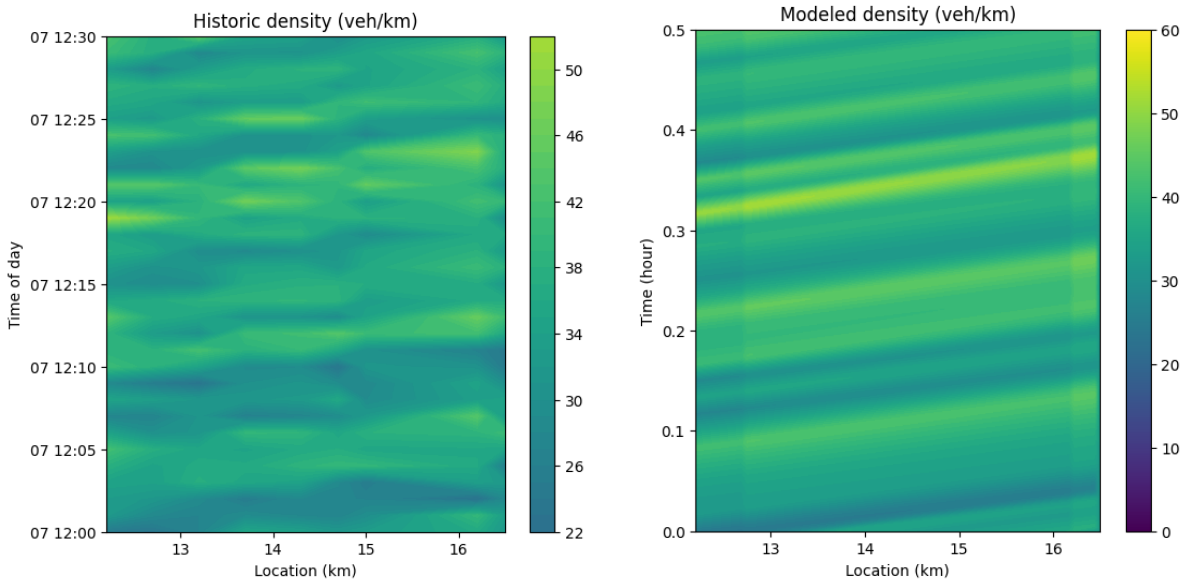


Figure 5.4: Historical density and modeled density plots of traffic between Delft and Rotterdam on the A13 on November 7th between 12:00 and 12:30. In the historic density plot, it can be seen that there is no congestion happening in this domain. The modeled density also only shows freeflowing traffic.

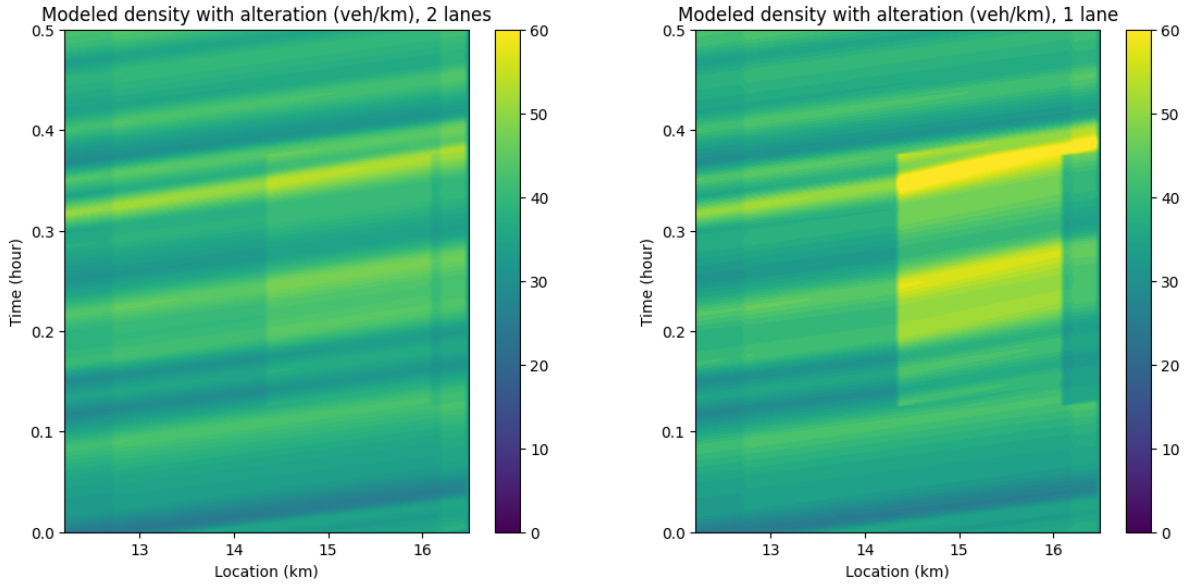


Figure 5.5: Two hypothetical modeled density plots of the highway between Delft and Rotterdam on the A13 on November 7th between 12:00 and 12:30. In both situations, we have added an obstruction on the highway where only two lanes are available (left plot) or only one lane is available for traffic (right plot). We can see that traffic drives slower when there is only one lane available.

6 Discussion

6.1. Discussing the results

This research includes the implementation of an artificial neural network, a numerical approximation of differential equations, and macroscopic traffic model at the basis. All of these sprockets work together to approximate something that is inherently random and uncertain: traffic. We have shown the predictive capabilities of our model in chapter 5, but even though the results look promising, we are missing performance indicators from similar researches. This will become a recurrent theme in this discussion

If we look at the results, we can distinguish multiple parts of the model. The first, and actually the innovative part of this master thesis, is the neural network that predicts the parameters of Smulders' FR. These parameters can then be used with historic traffic densities and a rewritten version of Smulders' FR to predict the flux of traffic. The neural network is able to learn from a large training set of historical traffic data. During the hyperparameter tuning experiments it has become clear that the learning process is robust, given a learning rate of 0.01 and 2 or 3 hidden layers.

In our experiments, we have trained the model only during 25 epochs. Considering that the training set has more than 1.5 million rows spread out into batches with size 1000, we still have more than 1500 learning steps per epoch. That is why the neural network also does not make much progress after 10 epochs. It can be said that the number of epochs could have been higher, especially in some hyperparameter tuning experiments. This could help showing the true quality of a predictive model with a lot more variables, e.g. a neural network with 4 or more hidden layers. This can also show us the point where overfitting does occur, as there was never a sign of overfitting in this research.

We have trained a neural network with 2 hidden layers and a learning rate of 0.01. In figure 5.1, we can see an R2 score of 0.7942 between the predicted and actual flux. This is decently high, especially considering the randomness of traffic. This means it is possible to predict the parameters of Smulders' FR, and thus the relation between density and flux, based on the MSI data.

The neural network is only part of the predictive traffic model. The other part consists of a traffic model, that uses the fundamental relation and the Godunov scheme to replicate the flow of traffic. In chapter 5, we have first shown the performance of this traffic model without any alterations added to the traffic situation. Afterwards, we have added different obstructions to see how the flow of traffic would respond.

One first conclusion that we can make from these experiments, is that the model replicates the historic traffic patterns decently well. We have seen that the model accurately depicts free-flowing traffic in its velocity. Specific patterns in congestion can however be more difficult to model, as it is seen in figure 4.10. Nevertheless, the shape and severity of congestion is consistently modeled realistically.

One flaw of the traffic model is the lack of a performance indicator, that could have been used to compare our model to other models. Without these indicators, we aren't able to compare our traffic model to any other model. We can still draw some conclusions about the performance of the traffic model when we just compare it to historical data. These have been discussed in the previous paragraph, and in chapter 5. The example cases in chapter 5 have all shown that the traffic model is able to replicate historic density plots based on MSI data.

Looking back at our initial goals, then we can see that mathematical models are able to improve the

estimates of the effect of road work on traffic. We have designed and trained a predictive model that is able to model alterations to the traffic situation and produce realistic results. However, we are still missing some performance metric to compare this method against, as we have mentioned before.

6.2. Opportunities and future research

It can be seen in the results (chapter 5) that the traffic model performs well for certain cases, but there are still some points of improvement. These improvements will not change the applied method much, but will definitely expand the possibilities of the model. We will discuss these opportunities in this section.

A first expansion to the predictive model would add other inputs to the neural network. The goal of this neural network is to predict the behaviour of traffic, i.e. the parameters of Smulders' FR. We assume that these parameters are only dependent on the traffic situation, but in reality there are a lot more factors that come into play like weather type, vehicle composition or time of day. The layout of the model will allow us to easily add these factors to the inputs of the neural network.

Another subject for future research lies within the traffic model itself. Our model is only able to replicate a highway without entries or exits. This choice was made to simplify the implementation of the Godunov scheme, because these models with accessways (on/off the highway, or to other highways) requires route-choice algorithms. However, if we want to research the full effect of road alterations, then the model should look at roads like a network instead of one path. Purely scaling the traffic model will not be a groundbreaking research, as there have been implementations of the LWR model in previous research that can model a full network, like in [4]. Still, the combination with our predictive model will definitely add meaningful insights.

After looking at modeled density plots with and without alterations, we have drawn the conclusion that the traffic model accurately replicates historic density plots, and that we can use this model to predict the effect of alterations on traffic. By alterations, we mean a change in the number of available lanes and a change in the maximum allowed velocity. We have found some opportunities that use these predictions.

First, we could use this traffic model to predict the effect of road work on traffic. The model can be used to approximate the increase in travel time during congestion caused by obstructions like a narrowing in the road, or a reduction in maximum allowed travelling speed. It can also be used to approximate the positive effect of road work, by modelling the traffic flow after roads and/or lanes are added. It is already possible to use our traffic model to create realistic density plots for historic situations with an alteration, but we are not yet able to calculate an increase in travel time.

Finally, this traffic model is relatively fast. With a few additions, it can be used as a lightweight traffic model with interactive abilities. Traffic managers could use it to experiment with the traffic situation by changing roads, or by changing the traffic intensity at will.

7

Conclusion

We have researched the impact of road alterations on traffic flow using a macroscopic traffic model in combination with a neural network. As a traffic model, we have used the LWR model with Smulders' fundamental relation. This specific model relies on the idea that the behaviour of traffic is locally defined by three parameters: u_0, q_j and q_c. We have trained an artificial neural network to predict these parameters based on the available lanes and the speed limit. While the application of neural networks in traffic modeling has been researched thoroughly [24], our methodology offers a new approach to predictions of the effect of road alterations.

By rewriting Smulders' fundamental relation, we were able to combine it with the artificial neural network to create one trainable predictive model. The dataset contains loopdata from Dutch highways, supplemented with historic data from overhead matrix signs. This data was retrieved through NDW, the Dutch autority on traffic data. The predictive model is able to predict the flux of traffic with an R2 score of 0.7942. We can conclude that the predictive model shows promising levels of accuracy.

The other part of the traffic model consists of a numerical approximation of traffic. We approximate the exact solution to the LWR model with the Godunov scheme, where the fundamental relation on every boudary is dependent on the MSI data on that point. This allows us to replicate the historic traffic density on a highway without entries or exits based on an initial density, the influx of vehicles and the road situation (i.e. the speed limit and the number of available lanes on every point along the highway). Figures 5.3 and 5.4 have shown a solid performance of the traffic model, however we have not been able to compare our approach to comparable studies. This is a shortcoming of this research.

Our novel approach to a variable fundamental relation allows us to modify the traffic situation, as it can be seen in figuress 5.3 and 5.5. They show promising results for the application of this model as a lightweight traffic model. We conclude that the effect of road work on traffic can be predicted with this apaproach. All of the used code can be found in `github.com/vossemeijssen/macroscopic_traffic_model`.

Overall, this study contributes to the body of knowledge about traffic modeling. Subjects for future research would be an extension of this traffic model from a single road to a network, and a more thorough analysis of the effect of alterations on travel times and congestion intensity. Another study could easily add more input variables to the neural network, which can improve the model and give even more parameters to modify in predictions.

References

[1] Nour Alqudah and Qussai Yaseen. “Machine Learning for Traffic Analysis: A Review”. In: *Procedia Computer Science* 170 (2020). The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops, pp. 911–916. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2020.03.111. URL: https://www.sciencedirect.com/science/article/pii/S1877050920305494.

[2] Jaume Barcelo. *Fundamentals of Traffic Simulation*. Jan. 2010. ISBN: 978-1-4419-6141-9. DOI: 10.1007/978-1-4419-6142-6.

[3] Y. Bengio and X. Glorot. “Understanding the difficulty of training deep feed forward neural networks”. In: *International Conference on Artificial Intelligence and Statistics* (Jan. 2010), pp. 249–256.

[4] Bjørnar Dolva Bergersen. “Numerical Solutions of Traffic Flow on Networks”. In: (2014).

[5] K. Fox. “Is micro-simulation a waste of time?” In: (2008).

[6] Sergei K. Godunov and I. Bohachevsky. “Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics”. In: *Matematičeskij sbornik* 47(89).3 (1959), pp. 271–306. URL: https://hal.science/hal-01620642.

[7] Victor L. Knoop Guopeng Li and J. W. C. van Lint. “Estimate the limit of predictability in short-term traffic forecasting: An entropy-based approach”. In: *Transportation Research Part C: Emerging Technologies* 138 (2022), p. 103607. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2022.103607. URL: https://www.sciencedirect.com/science/article/pii/S0968090X22000535.

[8] Dirk Helbing. “Derivation of a fundamental diagram for urban traffic flow”. In: *The European Physical Journal B* 70 (2009), pp. 229–241.

[9] Ansgar Jüngel. “Modeling and Numerical Approximation of Traffic Flow Problems”. In: *Lecture Notes (preliminary version)* (2002).

[10] Arash Khodadadi. *Traffic forecasting using graph neural networks and LSTM*. 2021. URL: https://keras.io/examples/timeseries/timeseries_traffic_forecasting/ (visited on 12/28/2021).

[11] Philippe Lepert and Francois Brillet. “The overall effects of road works on global warming gas emissions”. In: *Transportation Research Part D: Transport and Environment* 14.8 (2009), pp. 576–584.

[12] M. J. Lighthill and G. B. Whitham. “On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads”. In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 229.1178 (1955), pp. 317–345. ISSN: 00804630. URL: http://www.jstor.org/stable/99769 (visited on 07/12/2023).

[13] J.W.C. van Lint. *Data gedreven modelleren van verkeer en vervoer*. Delft, The Netherlands: Delft University of Technology, 2021.

[14] Ranju Mohan and Gitakrishnan Ramadurai. “State-of-the art of macroscopic traffic flow modelling”. In: *International Journal of Advances in Engineering Sciences and Applied Mathematics* 5 (Sept. 2013). DOI: 10.1007/s12572-013-0087-1.

[15] Loris Nanni, Andrea Loreggia, and Sheryl Brahnam. “Comparison of Different Methods for Building Ensembles of Convolutional Neural Networks”. In: *Electronics* 12 (Oct. 2023), p. 4428. DOI: 10.3390/electronics12214428.

[16] NDW. *NDW docs*. 2023. URL: https://docs.ndw.nu/ (visited on 01/23/2024).

[17] D. Ngoduy, S. P. Hoogendoorn, and Henk J. Van Zuylen. “Comparison of Numerical Schemes for Macroscopic Traffic Flow Models”. In: *Transportation Research Record* 1876.1 (2004), pp. 52–61. DOI: 10.3141/1876-06. eprint: https://doi.org/10.3141/1876-06. URL: https://doi.org/10.3141/1876-06.

[18] S. Paul and V. Knoop. *Traffic Flow Theory and Simulation*. Delft, The Netherlands: Delft University of Technology, 2016.

[19] Paul I Richards. “Shock waves on the highway”. In: *Operations research* 4.1 (1956), pp. 42–51.

[20] J. Ngoumou S. Heigl D. Hubber. *Godunov methods in GANDALF*. 2015. URL: https://gandalfcode.github.io/gandalf-school/talks/GodunovMethods.pdf (visited on 10/28/2015).

[21] David Tedjopurnomo et al. “A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges”. In: *IEEE Transactions on Knowledge and Data Engineering* PP (June 2020), pp. 1–1. DOI: 10.1109/TKDE.2020.3001195.

[22] Athanasios Theofilatos et al. “Meta-analysis of the effect of road work zones on crash occurrence”. In: *Accident Analysis & Prevention* 108 (2017), pp. 1–8.

[23] Femke van Wageningen-Kessels et al. “Genealogy of traffic flow models”. In: *EURO Journal on Transportation and Logistics* 4.4 (2015), pp. 445–473. ISSN: 2192-4376. DOI: https://doi.org/10.1007/s13676-014-0045-5. URL: https://www.sciencedirect.com/science/article/pii/S2192437620301114.

[24] Eleni I. Vlahogianni, Matthew G. Karlaftis, and John C. Golias. “Short-term traffic forecasting: Where we are and where we’re going”. In: *Transportation Research Part C: Emerging Technologies* 43 (2014). Special Issue on Short-term Traffic Flow Forecasting, pp. 3–19. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2014.01.005. URL: https://www.sciencedirect.com/science/article/pii/S0968090X14000096.

[25] Nationaal Dataportaal Wegverkeer. *NDW open data*. 2023. URL: https://opendata.ndw.nu/ (visited on 10/05/2023).

[26] Ira Winder and Nina Lutz. *Computational Ants: Agent Based Visualization Technique with CDR OD Matrix*. 2016. URL: https://ira.mit.edu/blog/agent-based-visualization (visited on 03/25/2024).

A

Rewriting Smulders' fundamental relation

In this appendix, we will show a full elaboration of the rewriting Smulders' FR, as can be found in chapter 2. First, we start with the formula for the flux of traffic according to Smulders' FR:

$$f_{\text{Sm}}(q) = \begin{cases} u_0 q - u_0 \frac{q^2}{q_j} & \text{for } q < q_c, \\ u_0 q_c - \frac{u_0 q_c q}{q_j} & \text{for } q \geq q_c. \end{cases}$$

We can rewrite both parts of the equation as follows.

$$\begin{aligned}
u_0 q - \frac{u_0 q^2}{q_j} &= u_0(q - q_c + q_c) - \frac{u_0 q(q - q_c + q_c)}{q_j} \\
&= u_0 q_c - u_0(q_c - q) - \frac{u_0 q q_c}{q_j} + \frac{u_0 q(q_c - q)}{q_j} \\
&= u_0 q_c - u_0(q_c - q) - \frac{u_0(q - q_c + q_c) q_c}{q_j} + \frac{u_0 q(q_c - q)}{q_j} \\
&= u_0 q_c - u_0(q_c - q) - \frac{u_0 q_c^2}{q_j} + \frac{u_0(q_c - q) q_c}{q_j} + \frac{u_0 q(q_c - q)}{q_j} \\
&= u_0 q_c - \frac{u_0 q_c^2}{q_j} + (-u_0 + \frac{u_0 q_c}{q_j} + \frac{u_0 q}{q_j})(q_c - q) \\
&= u_0 q_c - \frac{u_0 q_c^2}{q_j} + (-u_0 + \frac{u_0 q_c}{q_j} + \frac{u_0 q}{q_j}) ReLU(q_c - q) \text{ for } q \leq q_c \\
u_0 q_c - \frac{u_0 q_c q}{q_j} &= u_0 q_c - \frac{u_0 q_c(q - q_c + q_c)}{q_j} \\
&= u_0 q_c - \frac{u_0 q_c^2}{q_j} - \frac{u_0 q_c}{q_j}(q - q_c) \\
&= u_0 q_c - \frac{u_0 q_c^2}{q_j} - \frac{u_0 q_c}{q_j} ReLU(q - q_c) \text{ for } q \geq q_c
\end{aligned}$$

Using these results, we can see that $f_{\text{Sm}}(q)$ can be rewritten into one line.

$$\begin{aligned}
f_{\text{Sm}}(q) &= u_0 q_c - \frac{u_0 q_c^2}{q_j} + (-u_0 + \frac{u_0 q_c}{q_j} + \frac{u_0 q}{q_j}) ReLU(q_c - q) - \frac{u_0 q_c}{q_j} ReLU(q - q_c) \\
&= \frac{u_0}{q_j}(q_c q_j - q_c^2 + (q_c + q - q_j) ReLU(q_c - q) - q_c ReLU(q - q_c))
\end{aligned}$$