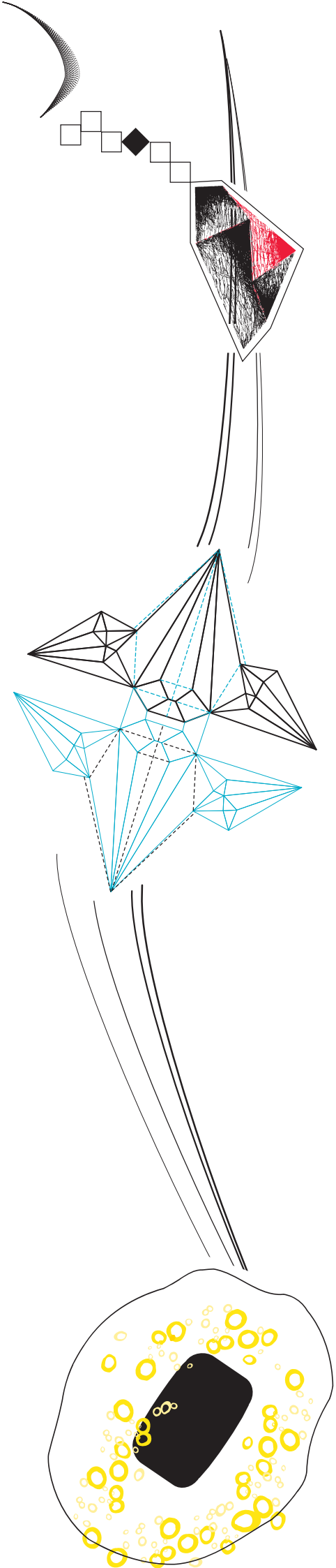


Using machine learning algorithms to improve traffic state estimation

A study on the usability of machine learning techniques in traffic state and speed estimation.

Master thesis

Antoon Dommerholt

August 29, 2019

UNIVERSITY OF TWENTE.

Colophon

Title:	Using machine learning algorithms to improve traffic state estimation
Subtitle:	A study on the usability of machine learning techniques in traffic state and speed estimation.
Date:	August 29, 2019
Author:	A.J. (Antoon) Dommerholt
Student number:	s1026534
Email:	antoondommerholt@gmail.com
Telephone:	06 465 88 456
Institution:	University of Twente
Faculty:	Engineering Technology (CTW)
Research group:	Centre for Transport Studies (VVR)
Master:	Civil Engineering and Management
Track:	Transport Engineering and Management
UT–supervisor:	Prof. Dr. Ir. E.C. (Eric) van Berkum
Supervisor:	Dr. Ir. L.J.J. (Luc) Wismans

Summary

Traffic state is defined by the traffic variables intensity, speed and density. When two of the three defining variables are known a traffic state can be determined. When only one of the variables is known, additional information is needed for traffic state estimation.

INWEVA is an overview of intensities on the Dutch national roads. For the parts of the road network that are not covered by detection loops, intensities are estimated by a model. Since only an estimation of intensities is known for these road sections, traffic state cannot be determined directly. Additional procedures have to be taken to estimate speed, and thus defining traffic state.

In this research the relation between intensities and speeds is studied. This research aims to give a good estimation of speed, based on intensity data only. When speeds are known, it can be determined whether or not congestion occurs, and traffic state is defined. The estimations of traffic state are made by inputting intensities into machine learning models. The main question for this research is given below and is answered by researching the appropriate machine learning technique for traffic estimation and researching if there are additional attributes that may improve estimation. The last subquestion tries to find an answer to whether or not characteristics of road section are transferable to other road sections, in order to train models on other road sections than they will be tested on.

How can machine learning techniques estimate traffic state based on intensity data?

Literature review

The relation between intensity and speed is given by the fundamental diagram (FIGURE 2.1). This fundamental diagram consists of two parts, a free flow branch and a congested branch. In both these branches the same intensities occur, the speed, however, is different. Because the same value for intensity can belong to different speeds, speed cannot be directly derived from intensities.

Different researches have tried to make estimations of traffic state, only having one defining variable available. This can be the case in speed estimation, only knowing intensities (in cases of single detection loop data), or in estimating intensities, only having speed data available (e.g. from floating car data). All methods that are doing these estimations need to have additional input. It is rather impossible to find a direct relation between the traffic state variables, based on only knowing one of them.

Many different techniques for machine learning are available. Two main categories in supervised machine learning are decision tree learning models (DTL) and artificial neural networks (ANN). Regression is a mathematical approach that also tries to find relation between the input and output variables. Machine learning models are trained on exemplary data and tested on another set for their performance. Recurrent machine learning models perform well at capturing patterns where time sequences are involved.

Methodology

Data from NDW (Nationale Databank Wegverkeersgegevens) is used for training the machine learning models. This NDW data, that consists of intensities and speeds, is used as input to the model. This one minute aggregate data is aggregated to a 15-minute resolution and prepared to be used by the machine learning software. A set of locations is selected for this research and this is researched over a timespan that covers many situations.

The estimation of traffic states consists of two parts: congestion estimation and speed estimation. The performance of congestion estimation is measured by the f-score, which combines precision and recall. For speed estimation the performance measure is the root mean squared error (RMSE), which gives an indication of how much the estimation is on average apart from the measured value.

Different types of machine learning models are tested, using the WEKA software, for their performance. Before testing machine learning models, regression models are tested. The inputs that are used in the model are only intensities. Intensities from the selected location and an upstream and downstream location. For these three locations the input is given for the time that it is estimated, as well as the intensities on the interval before it and after it. The different models are compared to each other using the performance measures.

Additional input attributes are tested for their influence to the model outcomes. The additional factors tested are influence of the weather, deviation in the intensity data and the percentage of long vehicles. The outcomes of the attribute testing is compared to the values for testing only on intensities.

The last part of this research consists of testing models that are trained on data from multiple road sections. In the first place a model is tested that was trained on a combined training set of all roads, including the training set of the road that is tested on. In the second place a model is tested that is trained on a dataset that does not contain instances of the road section that is tested. These results are compared to earlier results. In order to combine data sets, road specific characteristics were added, such as speed limit, distance to up– and downstream location, number of lanes and what kind of bottleneck is present. Also, the intensities are scaled to a percentage of the roads capacity, in order to make the intensities of the different roads comparable.

Data

The data that is used, is NDW data that comes from MONICA or MONIBAS detection loops. MONIBAS is data that is already further processed than MONICA data. MONICA, however, more often contains data on vehicle length, this is one of the reasons also often MONICA data is used. Whenever MONICA data is used, it is processed in the same way MONIBAS data is processed, so both sources can be used together. Not all data does contain information on vehicle classes, most of the locations that are selected do have information on this.

The NDW data comes in a resolution of one minute. The input for the model in this case is 15 minutes, so it has to be aggregated to this resolution. Also the different lanes on the road are aggregated. For the data that is used for congestion estimation, an instance is considered congested when the speed is below 70 km/h. From this data instances are formed that can be used as an input to WEKA.

A total of twelve locations are selected for this research. These locations have a variety for several road situations. Most of the selected locations contain a bottleneck. Six have a decrease in the number of lanes, four have an on-ramp and two locations do not have a bottleneck at the site. A total of nine weeks of data is used divided throughout the year. Three weeks are taken from October 2017, three weeks from January 2018, and three weeks from April 2018.

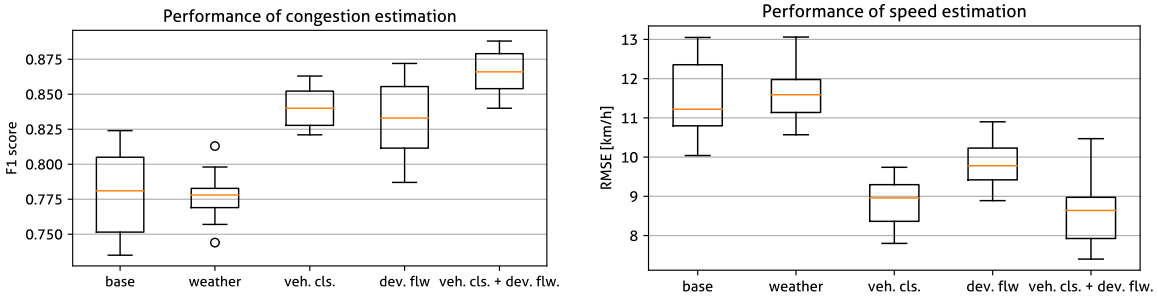
Results

Different kind of models are tested and scored using the performance measure. Testing has been done both in congestion estimation as in speed estimation. Testing these models took place at a road section on the A27. A summary of the results of this model testing is shown in TABLE 1. In all cases the recurrent version of the model performs better than the non-recurrent version. Recurrent neural networks (RNN) show both in congestion as in speed estimation the best results. A f-score of 0.90 is found for congestion estimation and a RMSE of 9.4 (km/h) for speed estimation. Because RNNs score highest, they are used for all other testing in this research.

Weather, vehicle classes and deviation within the intensity data have been tested as additional input attributes for their influence on the performance of the model. The road that is tested on is a section on the A58. This road section was chosen because this road section has information about vehicle classes available. In FIGURE 1 the results of this testing is shown. Information about weather does not lead to improvement of the model. Adding information on vehicle classes and deviation in the intensity data does improve the results significantly. Combine the latter two even gives slightly better results.

TABLE 1: Results of model testing on the A27.

Technique	Con. est. f-score	Speed est. RMSE
Regression / Logistic	0.61	26.2
Recurrent Regression / Logistic	0.84	23.7
J48	0.60	—
Recurrent J48	0.72	—
Random Forest	0.66	18.1
Recurrent Random Forest	0.69	15.8
Artificial neural network	0.83	19.4
Recurrent artificial neural network	0.90	9.4



(A) Congestion estimation

(B) Speed estimation

FIGURE 1: Results of adding additional input attributes in congestion and speed estimation on the A58.

Testing on a model that is trained on a training set that is combined from multiple road sections lead to scores that are comparable to the scores of models that are trained on a single road section. On average the results were slightly worse, but on average these changes are not significant. When a model that is trained on data, in which the road section on which is tested, is not included, results become very bad. There is hardly any estimation power left and these models should not be used for traffic state estimations. An overview of all results on all road is given in TABLE 2.

Conclusions

For traffic state estimation based on intensity data, the RNN is the most suitable machine learning technique. Because of the fact that traffic state is a temporal sequence, recurrent models are always preferred. The RNN is capable of estimating traffic state, based on intensity data. Best results are found in cases of clear bottlenecks, where all intensity data is available, for example at lane drops.

Weather input was found not to improve the results of the model significantly. Adding information on vehicle length and deviation in intensity data did result in a significant improvement in performance. A combination of the latter two resulted in even slightly better results. In congestion estimation the f-score improved by 7.8%, and in speed estimation the RMSE improved even 22.9%.

Having a model trained on a larger training set than the road section only, will lead to comparable results as when it is trained on a specific road section only. But when the training set does not contain instances from the road section that it is tested on, the model performs badly. This shows that the RNN is not capable of transferring the characteristics from one road section to the other. Neural networks seem not to be able to interpolate and extrapolate well.

Discussion

In the context of estimating speeds using intensity data of INWEVA, the results of this research cannot be used. In order to do so, a model should be trained on known data (from other road sections) as the use case. This research showed that doing that does not result in good estimations. For cases where it is

TABLE 2: Results of different models on all road sections. (The underlined values indicate the road and value this model was trained on.)

	Congestion Estimation (f-score)				Speed Estimation (RMSE)			
Road	RNN base[1]	RNN add. attr.[2]	RNN merged[3]	RNN other[4]	RNN base	RNN add. attr.	RNN merged	RNN other
A01_03	0.72	0.69	0.68	0.07	11.2	11.3	12.3	82.4
A02_02	0.66	0.66	0.72	0.50	9.7	9.2	10.1	13.8
A04_01	0.77	0.78	0.76	0.24	10.5	11.8	11.2	29.6
A04_02	0.86	0.92	0.93	0.42	8.8	6.8	7.5	51.5
A07_01	0.62	0.64	0.56	0.16	9.4	9.2	11.2	19.0
A27_02	0.90	0.89	0.85	0.66	9.4	10.5	11.3	39.6
A27_03	0.39	0.52	0.48	0.18	13.5	13.1	13.3	54.3
A28_01	0.48	0.50	0.51	0.20	15.1	14.4	14.8	22.0
A28_02	0.58	0.23	0.26	0.04	6.8	6.2	8.1	31.1
A58_01	0.81	0.84	0.87	0.26	10.4	9.7	11.3	44.7
A58_02	0.78	0.87	0.74	0.51	11.5	8.2	12.2	16.5
A58_03	0.87	0.89	0.86	0.53	10.0	9.2	9.8	14.6

possible to have a base measurement of intensities and speeds, this research shows, however, that for situations on that same road section where only intensities are known, it is possible to make proper estimations of the speed.

The methodology and the data have caused some limitations to the scope of this research. A lack of data caused that situations with ramps had incomplete intensity data, because many ramps were not measured. A limitation from the methodology is the limited number of locations that were chosen. By choosing more road sections, the dataset could have been a better representation of the Dutch motorway network. A limitation for the practical use is that data from the future time steps are used for estimation, this makes it impossible to apply the model real time.

There are a few directions for future research that can be followed from this research. The most interesting direction is carrying out the same research, but the other way round. Then a model would be made to estimate intensities, based on speeds, using machine learning techniques. Comparing those results to this research could give more insight in the relation between intensity and speed.

[1]Trained on intensity data only.
[2]Having added additional attributes.
[3]Instances from all roads are included in both the training set as the test set.
[4]Tested on a model that is trained on a model that does not contain instances from the same road as it is tested on.

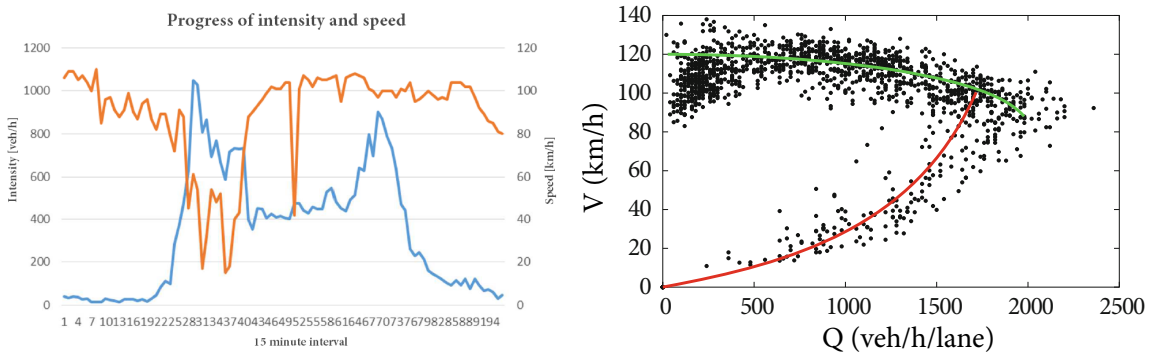
Contents

1 Introduction

In this introduction the context of the research is explained, the objective and the research questions are defined.

1.1 Context

Two elements that define a traffic state are the intensity on a road and the speed that is driven. Those two factors do not have a one-to-one relationship, which means that when only one of the two is known, the other cannot be derived from it. In FIGURE 1.1A the progress of both intensity and speed during a typical day are shown. What can be seen here is that a specific value for intensity does not always correspond to the same speed. This can be seen more clearly in the fundamental diagram in FIGURE 1.1B, where speed / intensity combinations are plotted. Because the same intensity can belong to a low speed (around the red line) or to a higher speed (around the green line), having knowledge about a specific value for intensity will give no certainty on the corresponding speed.



(A) Progress of intensity and speed

(B) The fundamental diagram

FIGURE 1.1: Example of the progress of intensity and speed (DatMobility, 2017) and a fundamental diagram (Treiber & Kesting, 2013), showing that the same intensity does not always correspond to the same speed.

In the Netherlands there is an overview of intensities on the Dutch national roads, which is called INWEVA (meaning: intensities on road stretches). For a significant part the intensities are measured by detection loops, but there also is a part of the network that is not covered by loops. For these uncovered road stretches the intensities are estimated by a model (Rijkswaterstaat, 2018). For the measured road stretches both intensities and speeds are known, since the loops that are used in the Netherlands are double loops, that can both measure intensity and speed. For the unmeasured road stretches all that is available is a modelled estimation of intensity.

As mentioned before, with only intensity data, speed cannot be determined evidently. This makes it also difficult to determine whether or not congestion takes place at a road stretch, with intensity data only. Being able to estimate or predict congestion state and speed could hugely improve the INWEVA data for the Dutch national roads. When speed were to be linked to the corresponding intensity data, a complete picture of traffic state for the national road system could be given. Also only being able to link congestion to the intensities would be of great benefit, since it can be identified where bottlenecks are located and when they are likely to be congested.

A research on identifying congestion based on intensity data was conducted by DatMobility (2017). The intensity data that was used was measured traffic intensity with 15 minute intervals. An artificial neural network was used to find out if it is possible to predict whether or not congestion is occurring, based on these modelled intensities. Some good results were found at locations where often congestions

occurred by clearly identifiable bottlenecks. There is, however, improvement possible on the method and the techniques used.

This research builds on the research done by DatMobility (2017). It has the same goal, which is identifying traffic state, based on 15 minute intensity data. Also machine learning techniques are used in order to achieve this goal, but more advanced techniques were applied to the problem. Also this research goes one step further in estimation of traffic state, since it does not only try to detect congestion, but does also try to give an indication of the speed at locations. Further this research is not limited to single road sections, but aims to make a framework that can estimate traffic state throughout the whole network of motorways in the Netherlands.

1.2 Research objective and research questions

Before giving an overview on how the research is conducted the objective and research questions are formulated. The aim of this research is using machine learning techniques to give an estimation on traffic state, based on intensity data. This will be useful for cases were only intensity data is known and more information is desirable, such as the modelled intensities on the parts of the Dutch national roads were traffic monitoring is not present. The formal aim of this research is generalized from this context and is as follows:

> Using machine learning techniques to give an estimation of congestion state and speed, based on intensity data.

This aim has been formulated as a research question that was attempted to answer in the research. To be able to answer this research question, it is divided into three subquestions, which together provide for an answer to the main question. The main question is defined as follows:

> How can machine learning techniques estimate traffic state based on intensity data?

The main question is divided into subquestions as follows:

1. What is the appropriate machine learning technique for estimating traffic state?
2. What input variables are important in the machine learning process?
3. How can a general approach be made for estimating congestion state and speed based on 15 minute aggregate intensity data on Dutch highways?

1.3 Thesis outline

This thesis starts with a literature review, which is focused on how traffic state is usually estimated and what variables play a role in it. Also different types of machine learning algorithms are discussed. A few examples of how machine learning can be combined with traffic engineering from literature are discussed.

In the methodology it is explained how this research is conducted. All steps that are taken are explained and motivated. The methodology is followed by a chapter on data collection. Choices for which data are used, is motivated. Also the process how the data is prepared is described.

In chapters 5 to 7 the results of the research are presented. This starts with testing different types of machine learning models, in which regression models, decision tree learning models and artificial neural networks are tested for its suitability in traffic state estimation for a particular road section. The best tested model is used for additional attribute testing. In the last phase of this research it is tested if the model still has estimating power when the model is trained on multiple road sections and when the model is trained on road sections that are not included in the test set.

In the conclusions the research questions are answered. In the discussion it is discussed how these findings can be of practical use in the outlined context. Also the limitations are discussed and recommendations for future work are given.

2 Theoretical framework

2.1 Traffic states

Traffic state is defined by the variables speed, flow, and density. The relation between these will be discussed in this chapter, by using the fundamental diagram. This is followed by a short view on how traffic state can be estimated, when not all variables are known.

2.1.1 Fundamental diagram

In highway traffic there are roughly two traffic regimes. In the free flow regime densities are low enough that congestion does not appear and single vehicles can to some extend choose their own speed. In this traffic regime speeds will generally be high. In the congested state speeds are lower than the free flow speed. The maximum intensity is found at the place where the congested regime and the free flow regime meet. This can be seen in FIGURE 2.1 at the place where the green and the red line intercept. In traffic state estimation all traffic variables that define the current traffic state need to be estimated (Wang & Papageorgiou, 2005).

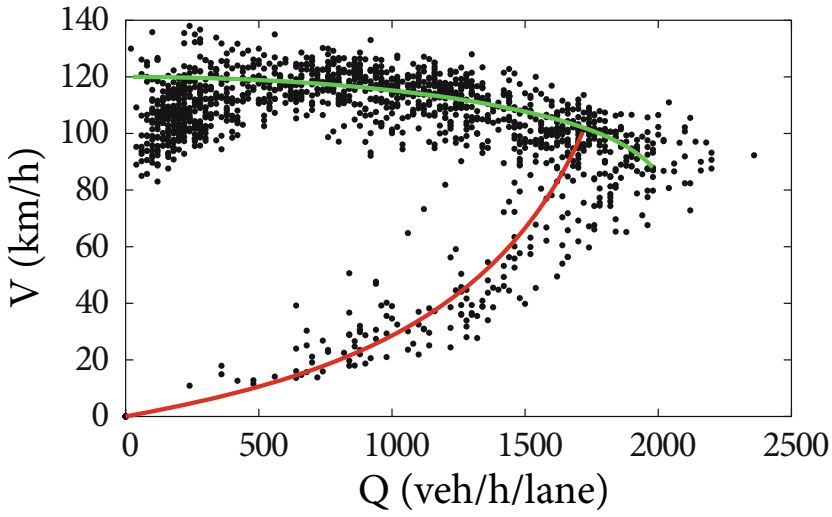


FIGURE 2.1: The relation between intensity and speed (one minute intervals) in the fundamental diagram (Treiber & Kesting, 2013).

The fundamental diagram of traffic describes the relation between the three variables that define traffic state in stationary conditions. Those variables are speed (v), density (ρ) and intensity or flow (q), that relate to each other $q = \rho v$. Theory of the fundamental diagram assumes that these three variables satisfy the EQUATIONS 2.1 and 2.2 (Seo, Bayen, Kusakabe, & Asakura, 2017).

$$v = V(\rho) \tag{2.1}$$

$$q = \rho V(\rho) = Q(\rho) \tag{2.2}$$

In which V and Q represent the fundamental relations between speed-density and flow-density respectively. This fundamental diagram plays an important role in traffic state estimation, because it is the core of traffic flow theory (Seo et al., 2017). Knowing this relation means that two of the three traffic state variables are needed to define a traffic state. The third variable can be deducted from the other two, assuming a stationary and homogeneous flow.

A visualisation of one of those fundamental relations is shown in FIGURE 2.1. Measurements of speed-intensity combinations are shown as black dots. In this relation between traffic intensity and speed two

different traffic states can be distinguished. The green line indicates the free flow condition, while the red line shows congested traffic flow. As can be seen in figure 2.1 a certain intensity does not always belong to the same speed. Roughly almost every intensity can belong to the free flow regime, or to the congested regime, with its corresponding speeds. This means that only measuring intensity will not automatically lead to knowing the speed or the current traffic state. Knowing the regime of the current traffic state would already give a much better view on the speed of traffic. Therefore separation of intensities in either the free flow or the congested regime would already be very beneficial.

Another way to determine traffic state when only intensity is known, is finding the density. Finding density, however, is difficult with the current technology. In order to find the density the number of vehicles on a certain road stretch in one moment needs to be determined. This is only possible when each vehicle is located at the same moment. Approaches of this can only be made when many vehicles would continuously transmit their location, which is not the case at the moment. Therefore, to estimate traffic state flow and speed are generally used.

What can be seen in figure 2.1 is that in the free flow traffic regime is that the green line is rather flat. This indicates that within in the free flow regime vehicles drive their maximum speed (speed limit), even when the intensity increases. Only with lower intensities in the free flow regime, the variation in speeds is higher, which is probably caused by freight traffic at night, which drive at lower speeds. The flow-speed dots in the fundamental diagram are aggregate values, so the speed is a average speed for a certain interval. This means that when the share of long vehicles (with a lower speed limit) in the interval is higher, the average speed will be lower, even though there are free flow conditions.

An interesting place in figure 2.1 is the place where the red and green line meet. This is the place with the maximum capacity during congested situations. In free flow conditions the capacity can be higher. When the traffic state changes from free flow conditions to congested conditions a capacity drop occurs (Treiber & Kesting, 2013). This means that as long as there is a congested situation, the free flow capacity cannot be reached. If the dots between consecutive flow-speed combinations were connected the effects of hysterics could be shown and it could be seen that the intensity needs to drop significantly during congested situations, before the system can recover again to free flow conditions. Because of these sequences, it is important to look at the history of traffic state when defining a current traffic state.

For the relation between intensity and speed, also the location and time play a role and the fundamental diagram function can be expressed as $Q(\rho, t, n, x)$, in which ρ is the density, t the time, n the type of vehicle and x the location (Seo et al., 2017). Therefore factors as these are important to consider when discussing the intensity-speed relation.

For the location it is important to know the road configuration, e.g. the number of lanes, whether it is located just before or after a bottleneck or if it is on a rather constant part of the road (no bottlenecks near and no changes in the number of lanes). For the time it is important to consider the traffic state of time intervals before the current time interval. The traffic state could come from a congested state, or from free flow conditions, or from a transition phase of evolving or resolving congestion.

2.1.2 Traffic state estimation

A traffic state is defined when two of the three defining variables (ρ, v, and q) are known. When this is not the case, and only one of the variables is known, techniques are needed to make a traffic state estimation. As described before density is a variable that is hard to measure, therefore to find a relation between traffic flow and speed would help increase the quality of traffic estimation

In uncongested conditions there are functions available to say something about the relation between flow and speed. The most well known is the function of the Bureau of Public Roads (BPR function) (Irawan, 2010). This function is shown in equation 2.3, where T is the travel time, T_0 the free flow travel time, v the flow and c the capacity. α and β are parameters. This function is proven to give a good estimate on travel times (which can be considered average speeds). In congested conditions however, where the flow that attempts to use a link exceeds the link's capacity, the output of this function becomes

unreliable.

$$T = T_0 \left(1 + \alpha \left(\frac{v}{c}\right)^{\beta}\right) \tag{2.3}$$

Instead of finding a direct relation between flow and speed, finding a relation between speed variance and flow could be an intermediate step. Blandin, Salam, and Bayen (2011) have researched this relation and found results as displayed in FIGURE 2.2. In general it was found that higher flows lead to lower speed variances. These speed variances are measured per individual vehicle, under stationary conditions. In the research of Bulteau, Leblanc, Blandin, and Bayen (2012), however, a positive relation between those two variables was found. This positive relation is caused by the differences in speed between lanes. Those findings mean that under certain conditions, it is possible to say something about flow, when knowing the variance in speed and vice versa .

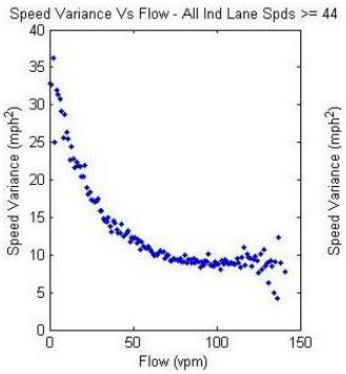


FIGURE 2.2: Relation between flow and speed variance (Blandin et al., 2011)

A better relation was found by Blandin et al. (2011) between speed and flow directly. Some examples are shown in FIGURE 2.3. Certain relations between the two variables can be seen here, in the form the fundamental diagram also has (FIGURE 2.1). But this form is not always the same and often it is unclear where the separation between the free flow and the congested regime is situated. The duality of some of the intensities (it is unclear whether an intensity belongs to the free flow or the congested regime) is not fully clarified by these relations.

Making estimations of speed, based on flow data have often been carried out by researches that do this based on single loop detectors (Coifman, 2001; Coifman & Kim, 2009). Making assumptions on or calculations of vehicle lengths is the base of these estimations. Jain and Coifman (2003) have conducted a research that is not based on this fact, but uses traffic flow theory principles to identify erroneous speed estimates. In this research data from adjacent lanes is combined to improve and filter bad estimations. This led to a significant improvement in speed estimations.

Also research was done on estimating flow, based on knowledge of speeds. Altintasi, Tuydes-Yaman, and Tuncay (2017) find that even though having only average travel speed information, it is still possible to detect critical patterns on urban roads. Those patterns were not real flows, but it were states as 'free flow', or 'dissolving congestion' etc. Seo et al. (2017) state that without additional assumptions it is not possible to derive traffic state from only knowing speed (from floating car data). Always a fundamental diagram is needed for estimation, that needs to be calibrated on stationary data (Seo et al., 2017).

Those researches show that under conditions it is possible to make estimations of speed based on flow data and vice versa. A very clear relation between those two traffic state defining variables, however, cannot be shown, there is always additional data needed to say something about its relation. The main difficulty in estimation is caused by the duality from the fundamental diagram in the speed flow relation.

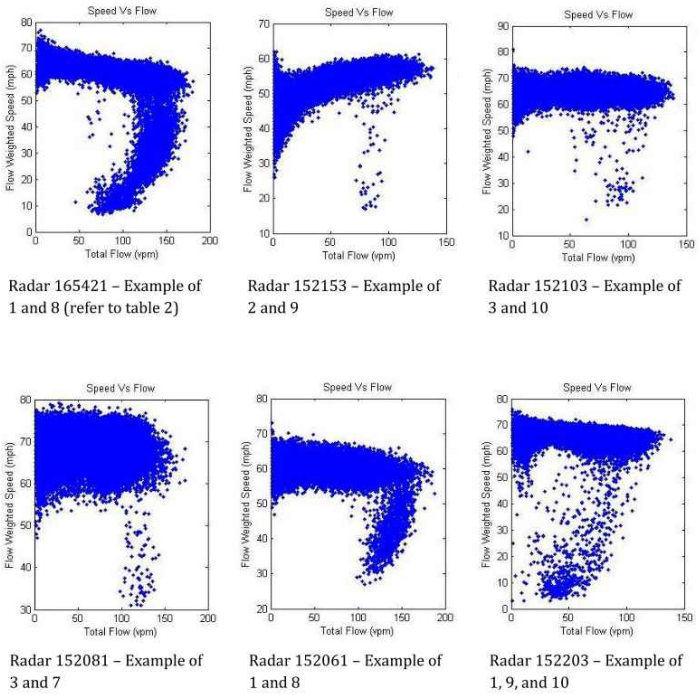


FIGURE 2.3: Relation between flow and speed (Blandin et al., 2011)

2.2 Machine learning in traffic engineering

Machine learning algorithms can be used for numerous different tasks. The most well known example for this is image recognition. In this case an image is shown to the computer and the computer will recognise the object. But its applications are much broader than image recognition. In this chapter some well known machine learning techniques are discussed. Also the application of machine learning in the field of traffic engineering is discussed.

2.2.1 Regression

Before discussing machine learning techniques, the principles of regression are discussed, since regression is more or less a basis of machine learning. In regression, input variables are taken to predict an output variable, based on known examples (a training set). In contrast to machine learning, which is a black box, regression uses mathematical functions in order to make a proper prediction on a certain output variable.

Regression works by finding a function $h_\theta(x)$ that comes closest to the values y of the training set. When y is only dependent on one variable x, a cost function J can be defined, based on the function $h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$. This cost function for m samples is given in EQUATION 2.4 (Ng, 2018). This cost function equals the mean squared error (MSE) between the estimated set and the set of the real values. Minimization of the cost function $J(\theta_0, \theta_1)$ will find the values for θ that makes the best possible predictor for y, based on this regression technique. The cost function can be minimized using the method of gradient descent, which is discussed in CHAPTER 2.2.3 on artificial neural networks.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{2.4}$$

Often y is not dependent on only one variable, but there are multiple variables involved. The function $h_\theta(x)$ that describes y in this case can be written as in EQUATION 2.5 (Ng, 2018), in which $x_0 = 1$ for the convenience of writing it vectorized. To find the values for θ the algorithm in EQUATION 2.6 can be used for all features j of the regression model. In this algorithm α is the step size for the convergence of the model (Ng, 2018).

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$
$$= \begin{bmatrix}\theta_0 & \theta_1 & \cdots & \theta_n\end{bmatrix} \begin{bmatrix}x_0 \\ x_1 \\ \vdots \\ x_n\end{bmatrix} = \theta^T x \tag{2.5}$$

repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \qquad \text{for j := 0 ... n} \tag{2.6}$$

}

The difference between regression and other machine learning techniques is that with regression the classification of data is based on mathematical rules and formulas. While with machine learning, the algorithm learns by example, instead of by rules. This makes that regression can be used for problems in which correlation can be easily found. In other cases, for example when it is not clear why a certain correlation is caused, or when the most accurate prediction possible are desired, machine learning techniques are a good alternative (Stewart, 2019).

2.2.2 Decision tree algorithms

A decision tree is a model for supervised learning, in which the local region is identified in a sequence of splits (Alpaydin, 2010). The model is made of decision nodes in which a certain property is tested and dependent on the outcome a different path is chosen. By doing this all data is split into categories that will be identified by going through all steps. A simple example can be seen in FIGURE 2.4. For each decision that is to be made a boundary needs to be found between two classifications. The main strategy that is followed by a decision tree algorithm is 'divide-and-conquer', this means that the problem is divided by the nodes in the tree, and it is attempted to make the division in each step as big as possible, in order to reduce the amount of decision nodes.

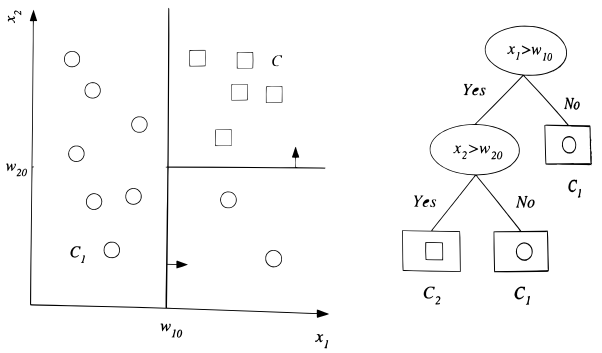


FIGURE 2.4: Decision tree in which the boundaries are shown as lines and the classifications as shapes (Alpaydin, 2010).

Decision trees are often used for classification problems. Advantages of decision tree models to more complex models, which may be more accurate, is that the model is very interpretable. The model can be written as a set of if-then rules and can be relatively easily understood by humans with knowledge in the field of application (Alpaydin, 2010).

2.2.3 Artificial Neural networks (ANN)

A machine learning algorithm that is widely studied is the artificial neural network. Those neural networks consist of at least three components, which are the inputs, the hidden units and the outputs (Nielsen, 2017). Those three components can be regarded as layers of units, in which all units from each layer communicate with all units from all adjacent layers. In FIGURE 2.5 a basic neural network is shown.

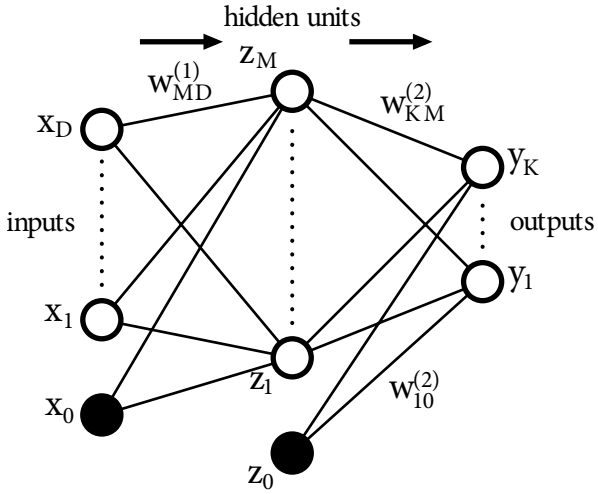


FIGURE 2.5: A simple neural network with one hidden layer (Bishop, 2006).

In a neural network, the nodes are called the neurons and can take any value between 0 and 1. All connections between the layers are weighted and so the value of the neurons in the next layer can be determined. The output of the neuron is determined by an activation function, often the sigmoid function (as can be seen in EQUATION 2.7) is used for this. The z in this function is a linear combination of all inputs and a bias and can be denoted as $\sum_j W_j X_j$, in which X represent the inputs (and the bias X_0, which always takes a value of 1) and W the weights (Nielsen, 2017). The reason that the sigmoid function is applied is to scale the output to values between 0 and 1. Another scaling function that is often used is tanh (z) which scales between -1 and 1.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.7}$$

As stated, the values for z are determined by a linear combination. An example of this in matrix notation can be seen in EQUATION 2.8 for the first step from the inputs $X_1 ... X_D$ to the hidden layer $Z_1 ... Z_M$. A similar transition is made in the step between the hidden layer $Z_1 ... Z_M$ to the output layer $Y_1 ... Y_K$. In these neural networks the first layer of neurons represent the inputs of the neural network (which is shown as $X_1 ... X_D$ in FIGURE 2.5, X_0 denotes the bias). In image recognition for example, every neuron in this network could represent the darkness of one pixel on a scale from 0 to 1. But in traffic engineering other inputs features can be given, such as present or past intensities together with the properties of the road.

$$\begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_M \end{bmatrix} = \begin{bmatrix} W^{(1)}_{10} & W^{(1)}_{11} & \dots & W^{(1)}_{1D} \\ W^{(1)}_{20} & W^{(1)}_{21} & \dots & W^{(1)}_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ W^{(1)}_{M0} & W^{(1)}_{M1} & \dots & W^{(1)}_{MD} \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_D \end{bmatrix} \tag{2.8}$$

The output layer ($Y_1 ... Y_K$ in FIGURE 2.5) provides the information that is requested. In the situation of traffic state estimation using intensity data this would be either a binary classification such as 'free flow' or 'congested', but it could also be more specific such as a real speed, or a range within the speed is expected to be. The output layer can be seen as a vector, in which all entries represent a classification. The number this vector entry (output neuron) holds, can be seen as a probability that given the input this classification would be correct.

Between the input and the output layer, there is the hidden layer. This layer is called the hidden layer since it will never be completely clear what the values these neurons hold exactly mean. However, behaviour of single neurons can be studied and some logical patterns may be found. In this way it may be cleared up what the hidden layer 'thinks', when it is trained.

The properties of a neural network are defined by the weight matrices. Finding good values for these weights is done by training the network. In order to do this all weights are initiated randomly and after this gradually improved, until satisfactory weights are found. This is done by trying to minimize the error function. This error function gives the distance between the model output and the known answers from the training set. In EQUATION 2.9 an error function is given (Bishop, 2006) for a training set of size n. In this function $\boldsymbol{t}_n$ stands for the target vector.

$$E(\boldsymbol{w}) = \sum_{n=1}^{N} E_n(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} \|\boldsymbol{y}(\boldsymbol{x}_n, \boldsymbol{w}) - \boldsymbol{t}_n\|^2 \tag{2.9}$$

This error function is often minimized with the method of gradient descent (Nielsen, 2017). With this method the weights are updated in order to make $E(\boldsymbol{w})$ smaller. This updating of weights takes place as in EQUATION 2.10 (Bishop, 2006), in which τ denotes the iteration step and $\Delta\boldsymbol{w}^{(\tau)}$ a weight vector update, which in case of gradient descent is given by the gradient vector of the error function $\nabla E(\boldsymbol{w})$. η is the learning rate of the method.

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} + \Delta\boldsymbol{w}^{(\tau)} = \boldsymbol{w}^{(\tau)} - \eta \nabla E(\boldsymbol{w}^{(\tau)}) \tag{2.10}$$

To find the updated weights $\boldsymbol{w}^{(\tau+1)}$, $\nabla E(\boldsymbol{w})$ needs to be determined. For the network shown in FIGURE 2.5, it can be found that

$$\frac{\partial E_n}{\partial w_{kj}^{(2)}} = \delta_k z_j \text{ and } \frac{\partial E_n}{\partial w_{ji}^{(1)}} = \delta_j x_i \tag{2.11}$$

in which

$$\delta_k = y_k - t_k \text{ and } \delta_j = (1 - z_j^2) \sum_{k=1}^{K} w_{kj} \delta_k. \tag{2.12}$$

By iteratively updating the weights, by calculating the gradient of the error function, using the updated weights, a satisfactory low value for the error function may be found in a local minimum of the error function. It is hard, or often impossible, to claim whether or not a local minimum is the global minimum. Therefore it may be useful to repeat the procedure with different initial – random – weights, since different local minima may be found. Although the method of gradient descent is commonly used, also other optimization methods are available as the Levenberg-Marquardt method and the Genetic algorithm (Ma, Tao, Wang, Yu, & Wang, 2015).

The main differences between machine learning methods such as decision tree algorithms and neural networks is that within decision trees it is relatively simple to see what happens in the model, while a neural network is more a 'black box' model, in which the algorithm learns by non-visible iterative steps. As it is easy to see how the decisions which are made within a decision tree algorithm separates different classes, it may be very difficult to find a proper decision tree when patterns within the available data are hard to find.

In artificial neural networks, logical patterns do not need to be the input to the model in order to make a classification of the data. The neural network iteratively optimizes its result for the given training data. This makes it a good method for classifying data in which patterns cannot be found or are very hard to identify. However, sometimes ANNs give good results on training data, because it optimizes for this, but fails to do so on other data.

2.2.4 Deep neural networks (DNN)

The difference between a 'normal' neural network and a deep neural network is the amount of hidden layers. In a conventional neural network there is only one hidden layer and only the amount of neurons is variable. In a deep neural network there are multiple hidden layers. This can range from a few hidden layers to over a hundred (MathWorks, 2017). In FIGURE 2.6 the architecture of a deep neural network is shown.

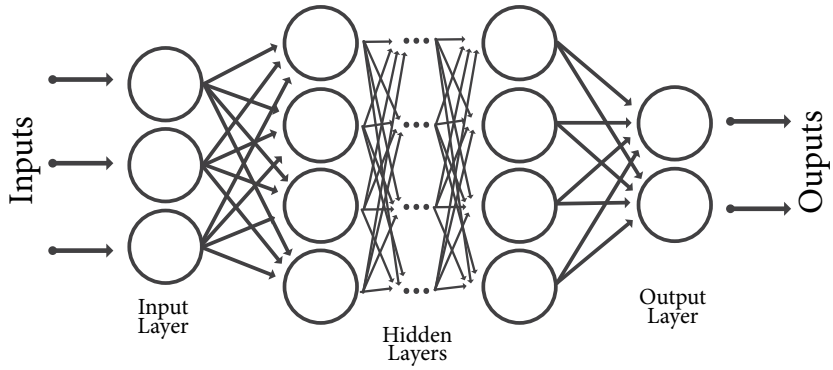


Figure 2.6: A deep neural network with multiple layers (MathWorks, 2017)

Advantages of deep learning networks compared to normal neural networks are improved accuracy of predictions made. Since there are many more connections to be made when there are more layers present. At the same time the large number of connections in a neural network has some disadvantages. In the first place it will take much longer to train the network. The fine-tuning of weights is the method to train the network, so with many more weights this fine-tuning will take much more time.

Since deep neural networks are more sophisticated than normal neural networks because of the higher amount of hidden layers, it can be trained so that for the training and the test data it will give very accurate results. However, this does not automatically mean that the network is of good quality. When the amount of connections is much higher than the amount really needed, the 'training' of the network will at some point become optimized for the provided data, instead of finding patterns that can be applied to other cases. So it will always be needed to evaluate the complexity of the network that is needed for the particular problem.

2.2.5 Recurrent neural networks (RNN)

Another variant on the neural network is the recurrent neural network (RNN). In this network the input is not only fed forward, but feedback loops are included, an example of such a network can be seen in figure 2.7.

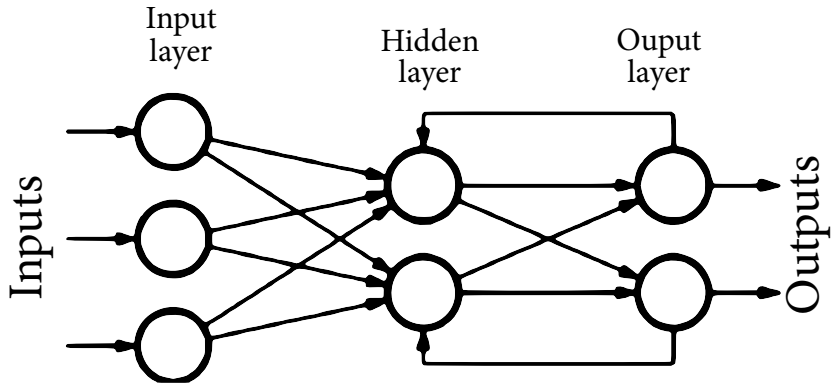


Figure 2.7: A recurrent neural network, in which the output layer is fed back to the hidden layer (Quiza & Davim, 2009)

In an ordinary neural network all inputs are sent through the network at once, so a single output is given. In a RNN time plays a role and the output is not determined instantly but with time steps. In every next step the output of the network will be fed back to the hidden layer. In this way with every next time step the output will change (Nielsen, 2017).

RNNs are good for the capturing evolution of traffic flow, volume and speed. Since RNNs use internal memory units for processing arbitrary sequences of inputs, RNNs have the capability of learning temporal sequence (Ma et al., 2015). Traffic patterns are defined by temporal sequences of the traffic

state variables, therefore RNNs are expected to capture patterns in a better way than ordinary ANNs.

2.2.6 Use for traffic state estimation

Several researches have been conducted on using big data / machine learning in traffic engineering. Most of these researches are about traffic flow prediction, but there are also some papers on traffic state estimation. Giving this short overview will help to see what techniques were found useful for which application in traffic engineering.

Most of the applications of machine learning within the field of traffic engineering are about forecasting the near feature, based on current traffic variables. Good results have been found by using (deep) neural networks for the short term prediction on traffic flow (Polson & Sokolov, 2017; Zhu, Cao, & Zhu, 2014) and using several machine learning techniques on short term prediction of traffic speed (Ma et al., 2015; Fusco, Colombaroni, & Isaenko, 2016). The challenge for this research, however, is not to make a forecast of traffic state, but to define traffic state when only one of the defining variables is available.

DatMobility (2017) has conducted a research on using an ANN to make an estimation of whether or not there is congestion, based on 15 minute aggregate data of modelled traffic flow. The input for this model intensity data was used. These intensities are from the road stretch for which the classification is made, as well from the road stretch upstream and downstream. The intensities that are used are the intensity of the moment for which the state of congestion is estimated and three data points of 15 minute aggregate intensity before that. So for the input intensities variation was made both in time and space. The research showed rather good results on the training set, but for other situations it mostly failed to detect congestion, so likely the ANN that was used was trained for the specific situation, in stead of being a generic model. This could have been caused by model overfitting, but it is more likely that there were characteristics of the specific road situations that were not included in the model. It is hard to say which characteristics this were exactly, but missing those makes that a model trained one road situation is not transferable to other locations.

2.2.7 Comparison of techniques

Regression is a non-machine learning mathematical approach for classifying data based on mathematical rules, while machine learning techniques learn by example and are therefore useful when a clear relation between input variables and the classification cannot be found. Within machine learning, two methods are decision tree learning and artificial neural networks. DTL divides the data into categories and by doing this a tree with branches and leaves is constructed. ANNs take input data and process it through hidden layers of neurons, in order to make a prediction of a specified category.

Variants on the ANN are the deep neural network and the recurrent neural network. The advantages of such more complex networks is that its results are often more accurate. RNNs can be useful when changes during time are important for the classification of data. Drawbacks of these more complex techniques are more calculation time and that it may become harder to correctly interpret the results.

When using DTL techniques it is very clear what the output of the machine learning is. It will be a set of rules on which decisions are made, that makes it useful for finding how the input variables are used within the model. RNNs are made for situations when patterns in time series are to be found, which is the case when classifying speeds base on intensity series. Because of the properties of these techniques it is chosen to apply these in the research.

3 Methodology

The approach that is used in order to be able to answer the research question is discussed in this chapter. Since the final aim of this research is to come up with a model that is able to estimate traffic state from intensity data, all steps that are taken will serve this goal. The following steps will be taken:

1. Collection of data;
2. Testing different kind of models;
3. Testing additional input attributes;
4. Testing multiple roads.

This methodology explains how those steps are taken.

3.1 Data collection

3.1.1 Data selection and gathering

In order to research the relation between intensity and speed patterns on highways using machine learning techniques, data needs to be acquired. This needs to be highway data where speed is linked to flow, in order to train a supervised machine learning model. Also this data should be diverse in terms of road properties, because otherwise the model could train for one specific situation. Within this data locations in which congestion occurs have to be included.

For the road stretches the data is acquired from the 'Nationale Databank Wegverkeergegevens' (NDW, 2018), which has a tool for downloading historical highway data on a one minute time resolution. The definition for road stretch will be derived from this NDW data. A road stretch or a road section is defined as the road around a detection loop. The beginning of this section is the upstream detection loop and the end the downstream detection loop.

Using NDW data means that the model is trained on measured intensities and not on modelled intensities, which is a use case for this research. But it is assumed that it is better to train a model based on measured intensity and measured speed data, than when modelled intensity data is linked to measured speeds.

For doing this research twelve locations are selected, that have a certain diversity on road characteristics. Most locations need to have an identifiable bottleneck, because these locations are expected to have the properties to make good estimations. Also locations without a bottleneck are included. All those locations need to have detection loop data available on its location, as well upstream and downstream from the location. For the researched location speed information must be available, for the upstream and downstream location intensity is enough.

In this research, complex road situations, such as weaving lanes at motorway intersections, peak lanes, motorways where transit traffic are separated from local traffic and other complex situations, other than lane drops and ramps, are not included in this research. Those situations are hard to fit in the model because of their specific characteristics. Using the format in this research where only intensities are used from the location (and upstream and downstream), would not be possible.

Testing the different types of machine learning models is done on a selected road section that has a very clear bottleneck, so good estimation can be made on the traffic state. Also for testing which additional properties, a location with a clear bottleneck is selected. This location must have information on vehicle length available, because this is one of the additional properties that the road is tested on.

For the selection of the road stretches a diversity is pursued. This means those road stretches will differ in the number of lanes, their place in the network, whether or not on– and off ramps are present, being upstream or downstream of a bottleneck and speed limit. This diversity will be in both the training and the testing data. For all road stretches both speed and intensity have to be known, as well for the researched road stretch as the road stretches up– an downstream. Not all this data can be acquired by NDW, but Google Maps is used to gain additional information.

Other properties of the researched road stretches that are needed are the speed limit at the time, whether or not an on– or off ramp is present, the number of lanes and for the up– and downstream road stretch, the distance to the researched road stretch and whether or not there is a lane drop. All those features are listed in TABLE 3.1. These properties are included because they are the main characteristics of the road. These properties will vary for each road section, so information on these properties is needed to identify the type of road. Having all properties that define a road situation included in the model may make it possible to use another road section for testing and for training.

TABLE 3.1: Input to machine learning model

Selected road stretch	Upstream road stretch	Downstream road stretch
Speed t_0	—	—
Intensity t_0, t_{-1}, t_{+1}	Intensity t_0, t_{-1}, t_{+1}	Intensity t_0, t_{-1}, t_{+1}
—	Distance to selected road stretch	Distance to selected road stretch
Speed limit	Speed limit	Speed limit
Number of lanes	Number of lanes	Number of lanes
On-ramp lane present	On-ramp lane present	On-ramp lane present
Off-ramp lane present	Off-ramp lane present	Off-ramp lane present
Presence of lane drop	Presence of lane drop	Presence of lane drop

3.1.2 Data preparation

Applying machine learning techniques will be done by using WEKA. WEKA is a software package that includes many different machine learning algorithms (University of Waikato, 2019). Algorithms that are included are regression methods, decision tree learning models and neural networks. WEKA is originally a Java based program that can be used using a GUI or in Java. There is a Python API available (Python-weka-wrapper, 2019), that is used in the research to include the WEKA models in the other program code.

The gathered data needs to be put in a form that it can be used by the WEKA software. But first the data needs to be aggregated in the way the data will be used. In order to be comparable to the research by DatMobility (2017), and to be applicable for comparable causes, it is chosen to aggregate the data to 15 minute data, because the modelled data that this research will be used for, also has a time resolution of 15 minutes.

Since NDW highway data comes with a resolution of 1 minute, this has to be changed. For intensities this is relatively easy, since all intensities can be added to each other. The best way for aggregating speed is calculating the space mean speed. This is not as straightforward as calculating the time mean speed, which is just a weighted average of all the speeds of every one minute data interval of the vehicles that pass. A way for finding the space mean speed is shown in EQUATION 3.1 (Soriguera & Robusté, 2011). This will be applied in aggregating the speed data from one minute to 15 minute intervals. It has to be noted that the original one minute aggregate data consists of time mean speeds, this cannot be changed to space mean speeds. This means that the 15 minute aggregate speed value is not the true space mean speed, but an approximation of the space mean speed.

$$\overline{v_s} = \overline{v_t} - \frac{\sigma_t^2}{\overline{v_t}} \tag{3.1}$$

Whenever an erroneous data entry is found, it is removed from the dataset if it is possible to still keep reliable data. This means that when for one aggregated 15 minute data point for example one minute of a certain vehicle category is missing, the data point can still be determined in a reliable way, since most data is available. In the case many data are missing or corrupted it is considered to delete that certain day for that certain road stretch from the dataset, since it may influence the model too much. If this is the case another day will replace this day of data, this will be a day that is the same day of the week and close to the original day that was used in the model. If structural errors or missing data is found for a certain road stretch, it is decided to replace this road stretch with a similar road stretch, that has no structural errors in the data.

3.2 Testing several models

When the data is prepared in the format that is needed for WEKA, the models are trained. It is chosen to train both a decision tree learning model (DTL) and a recurrent neural network (RNN). But before training these machine learning models a regression model is applied, in order to see the performance of linear regression.

The DTL model can be useful, because it can give a good insight in what happens in the model, because this will divide characteristics into branches and leaves. Also it has not been tested yet for this particular problem. RNNs are good in working with time series, and will probably give in this situation better results than ANNs, which are tested in the research of DatMobility (2017).

Since these models are only applied to one single road stretch, the speed limit, number of lanes and the prevalence of ramps will be constant on this road section, therefore they will not be included as an input to the model. The input that remains is the intensity data of the researched road stretch, the intensities up– and downstream and the distances to this up– and downstream locations. For testing the additional input attributes, these attributes are added to the intensities.

For congestion estimation multiple performance measures can be used. The most simple measure is the accuracy, which simply gives the percentage of correctly identified instances. A disadvantage of this measure is that in cases with a low number of congested instances, identifying all instances as not congested will lead to a very high accuracy. That is why it is chosen not to use accuracy as a performance measure. The measure that is used in this research is the f-score, which can be seen in EQUATION 3.2. The f-score is a compromise between precision and recall. The precision is the share of correctly identified cases of congestion of the total number of congestion identifications. The recall is the share of correctly identified cases of congestion of the total number of cases of congestion. These two measures include the number of false positives and false negatives. The f-score gives a better view on whether or not the model is able to capture congestion than precision and recall separately.

$$\mathrm{F}_1 = \left(\frac{\mathrm{precision}^{-1} + \mathrm{recall}^{-1}}{2} \right)^{-1} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{3.2}$$

For estimating speed the root mean squared error (RMSE) is the performance measure that is used (EQUATION 3.3, in which $\hat{Y}$ are the estimated speed values and Y the measured speeds). Other measures are available, such as the MASE (mean absolute scaled error), which just calculates the average of all errors. The advantage of the RMSE is that it gives a larger penalty to larger errors. This is useful in the researched situation, since the data can be roughly divided into congestion and free flow and having a larger penalty for larger errors, gives a big punishment for estimated speeds that are free flow and are estimated congested and vice versa.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2} \tag{3.3}$$

The model that is found to have the best results is used for the other parts of this research. Because it is expected that this model will also be best fitted for all other uses of the model, such as attribute testing. The best model will be validated by testing on the other road sections.

3.3 Testing additional attributes

In order to improve the results of the testing on a single road stretch, the influence of additional attributes is tested. These attributes are added to the intensities that originally were the only input to the model. A road section is selected for carrying out those tests. The following three attributes are tested:

1. Weather information;
2. Percentage of long vehicles;
3. Deviation in the intensity data.

For information on the weather KNMI data is used from a nearby station. KNMI has data on whether or not bad weather circumstances have occurred during a certain period. For most locations there is information available on the share of long vehicles, this data can be acquired from the NDW data. The last value that is tested is the deviation in intensity data. Because all data was aggregated to 15 minute intervals from one minute intervals, it is possible to calculate the standard deviation of the intensity of these intervals. This deviation can be an indication of changing circumstances.

Whether or not these three factor have influence is tested by comparing the f-scores and RMSEs to the base test, where only intensities are fed to the model. If a positive change is registered, this additional attribute can be of extra value to the model. The attributes that contribute positively to the model are also combined and tested. The best configuration found will be validated on the other road sections.

3.4 Testing on a set of multiple roads

At first the machine learning models were trained for the researched road stretches only, this means all selected road stretches are trained individually and tested individually. Now a general classification model will be made. In contrary to the previous part of the research, where only one road stretch was researched, now different road stretches together will serve as an input to the model. This means that also a variety in road properties is introduced at this point. So from this point onwards the speed limit, the amount of lanes and the prevalence of ramps are used as an input. Also intensities are normalized, in order to make them comparable to each other. This has been done by representing intensities as a percentage of the roads capacity. This capacity is determined by the value in which 99% of all values on that road are lower than the capacity value.

This part of the research consists of two parts. In the first part all road sections are combined in one training set and tested on each road individually. This makes it possible to see if adding multiple roads as an input confuses the model, or whether it is still possible to make proper estimations. The results of this testing are compared to the results of testing on models that are trained on one specific road.

In the second part the models will be tested on roads that are not included in the train set. This must show if it is possible that properties of the roads are transferable to other roads, so still good results can be found. Also here the finding are compared to the results of specific road testing.

4 Data collection

The data that is used to train a speed estimation model contains data of both traffic speed and intensity. In this chapter it is discussed which data is obtained, how it is processed and what are the properties of the data.

4.1 Data Sources

Two main sources for traffic analysis are available, which are loop detector data and floating car data. Since for floating car data a good estimation of speed can be made, but it is very hard to give a good indication of traffic flow, detector data is the preferable data source to use for this research and this is used for this research.

4.1.1 MoniCa / MoniBas

In the Netherlands many highways are equipped with the MoniCa detector system, which is a double loop detector in the road that is able to measure both speed and intensities. MoniBas is the name for the processed data from MoniCa. MoniBas checks the MoniCa data for reliability and missing data (De Jong, 2012).

NDW (Nationale Databank Wegverkeergegevens) is a corporation of several administrations in the Netherlands, such as provinces and Rijkswaterstaat. NDW gathers, manages and distributes traffic data in the Netherlands, that can be used for traffic information and traffic studies. It also makes detector data for the roads in the Netherlands available. MoniBas data, and also raw MoniCa data at some places is available in this data set. This dataset is the data source that is used for this research.

FIGURE 4.1: MoniCa and MoniBas locations where the data is divided into vehicle classes.

4.1.2 Characteristics of the data

All MoniCa and MoniBas data from NDW contain information on both intensities and speeds. This is available for each individual lane in one minute intervals. Some of the MoniBas data makes a distinction between different vehicle classes, also the raw MoniCa data does this. All passing vehicles are divided into three length categories (which can be found in TABLE 4.4).

This distinction can be used to see differences in speed between different vehicle classes or can be used to determine the percentage of long vehicles. This share of long vehicles is a property that is researched. Not all MoniBas data makes this vehicle length distinction, many data entries only provide the class 'anyVehicle', in which all lengths are aggregated. In FIGURE 4.1 it is shown at which locations in the Netherlands MoniCa and MoniBas data is available that have the different vehicle classes, these are the best locations for using in this research.

4.1.3 Data completion

When a minute of data is missing or corrupt, MoniBas completes the data if the duration of the missing data is not more than 5 minutes. Within the MoniBas system there is a check for whether or not data is correct. Sometimes the vehicle categories are only available in the MoniCa data and not in the MoniBas data. In this case MoniCa data is used and processed in a way that is is usable, such as the MoniBas data.

Whenever there is corrupt of missing data, a gap within the time line of data points evolves. MoniBas does fill in the values of this gap, if the length of the gap is not more than 5 minutes. For completing the data the last accepted minute value (t_1) from before the gap of missing data and the first accepted minute value (t_2) from after the gap are used. For the minutes in between (never more than 5), the value for intensity (I) or the speed (here defined as $1/v$) are chosen on the linear interval between the values of t_1 and t_2, as can be seen in FIGURE 4.2 and in EQUATIONS 4.1 and 4.2 (NDW, 2013). Whenever MoniCa data is used and data is missing, the same rules as for completion of the MoniBas data are applied.

$$I_t = I_{t_1} + (t - t_1)\frac{I_{t_2} - I_{t_1}}{t_2 - t_1} \tag{4.1}$$

$$\frac{1}{v_t} = \frac{1}{v_{t_1}} + (t - t_1)\frac{1/v_{t_2} - 1/v_{t_1}}{t_2 - t_1} \tag{4.2}$$

In which I_t is the interpolated value for intensity and v_t the interpolated value for speed. t_1 indicates the time of the last known value before the gap and t_2 the time of the first correct value after the gap. This interpolation is applied for all minutes within the gap, making the time series complete again.

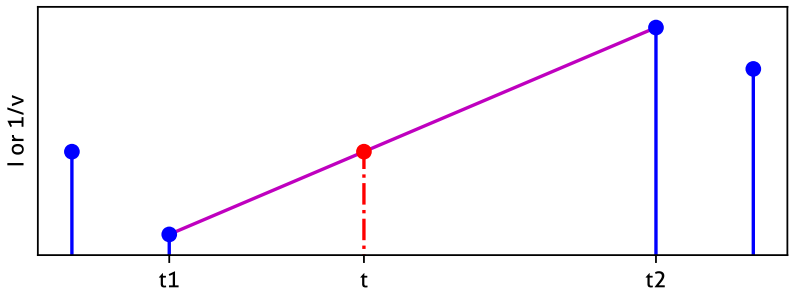


FIGURE 4.2: Completion of missing data in MoniBas. The blue points indicate normal data points in the dataset, the gap between t_1 and t_2 that came from missing or corrupt data is filled with interpolation, based on the last value before and the first value after the gap. When the gap is no longer than 5 minutes, this results into a continuing stream of data points.

4.2 Locations

A total of twelve locations have been selected for this research. These locations cover different kind of road situations. Most of these road situations are bottlenecks, but also situations without bottlenecks are researched. Different kind of road section have been researched, because the goal is to create a model that can estimate speeds on a wide range of locations

The reason that many bottlenecks are researched, is that at these locations the relation between intensity and speed is expected to be most easily demonstrated. At a bottleneck it is most times very clear what is the cause of congestion and congestion will occur regularly. Especially lane drops are bottlenecks that can well be fit into models, because in contrary to ramps, all traffic stays on the same road, there is no traffic leaving or entering the road.

In table 4.1 an overview is given of the locations, which are put on a map in figure 4.3. As can be seen six of the selected location contain a lane drop. four have an on-ramp and two have no particular bottlenecks. Most locations have information available on the length of the vehicles, just two do not. One location on the A27 near Breda is located on a bridge, which causes the lanes to be more narrow than normal lanes, which may be regarded as a flow preserving bottleneck. Another case that is unique in these locations is the A58 near Breda, where two motorways are merging. This can be seen as an ordinary lane drop, but upstream there is not one road, but there are two different roads.

Table 4.1: Specifications of the locations in this research.

location	# lanes	speed limit	on ramp	off ramp	lane drop	length info	bottleneck
A01_03	3	130	no	no	yes	yes	lane drop
A02_02	2	130	yes	yes	no	yes	on ramp
A04_01	3	100/130	no	no	yes	no	lane drop
A04_02	2	100	yes	no	no	yes	on ramp
A07_01	2	130	no	no	no	yes	none
A27_02	3	130	no	no	yes	no	lane drop
A27_03	2	130	yes	no	no	yes	on ramp / narrow lanes
A28_01	2	120/130	yes	yes	no	yes	on ramp
A28_02	3	120	no	no	no	yes	none
A58_01	3	120	no	yes	yes	yes	lane drop
A58_02	3	130	no	no	yes	yes	lane drop
A58_03	3	130	no	no	yes	yes	lane drop / merging highways

4.3 Time periods

Three time periods of three weeks are selected to use as input data for the model. These three periods are chosen throughout the year, so not only one single season is researched. From all three periods the first two weeks will serve as training data for the model and the last week as testing data, so all periods have both training and testing data. In table 4.2 the three periods are shown. Taking this time span of nine weeks creates 6048 instances per road of which two third is used for training, the other third for testing purposes.

For two days (5 and 10 October 2017) there was much missing data for all points. Therefore it was chosen to replace these days with two other comparable days. Both days were replaced by the same day of the week of the week after the October period.

A day that is included in the data set, that was remarkable, is 18 January 2018. On that day on many measurement locations low speeds were registered. This was caused by a severe storm in the Netherlands,

FIGURE 4.3: The twelve locations that are used in this research.

TABLE 4.2: Time periods in which is measured

	training set		**test set**	
	begin	end	begin	end
1	01-10-2017	14-10-2017	15-10-2017	21-10-2017
2	07-01-2018	20-01-2018	21-01-2018	27-01-2018
3	01-04-2018	14-04-2018	15-04-2018	21-04-2018

in which was advised not to drive (code red). In the chosen road sections however, no special congestion due to this fact was found and therefore it is chosen to keep this day in the dataset.

It is chosen not to exclude non-recurring traffic patterns. It may be argued that having only regular situation in the dataset makes better predictions possible, but excluding this data makes that this model can only identify recurrent patterns. When this data is included, the model can be trained on situations that do not often occur. Having this data included in the data increases its estimating power. Also it will probably not lead to confusion in the training set, as there will be enough examples of recurring congestion. This means that not excluding special event data will lead to the same capabilities as when they would be excluded, but including makes that the model has a higher change of also capturing special event situations.

4.4 Obtaining and processing the data

The data is obtained by making requests at NDW (Nationale Databank Wegverkeergegevens) for the selected time periods. The data was made available in large XML-files that contain all measurement points of the Netherlands. These XML files were parsed using a Python script and the selected locations – together with some upstream and downstream measurement points – were filtered out of this data and stored to a plain text format, which makes is easy to access the data when needed.

Since the NDW data comes at a time resolution of one minute, it was aggregated to 15 minute intervals, because that time resolution was chosen for this research. This was also done using a python script. For aggregating the intensity the values of all lanes and all separate minutes were added. This value is corrected, so the unit of intensity will remain veh/h.

For aggregating the speeds the harmonic mean is used in order to get an approximation of the space mean speed. Unfortunately the values that are provided by MoniBas are the arithmetic mean for one minute for one lane. Since this is the only value provided (no standard deviation), only the time mean speed on a one-minute scale is available. For the aggregation of the speeds the harmonic mean of all values is determined, so that the speed that is used can be seen as an approximation of the space mean speed.

It appeared that some minutes of data were missing in the raw MONICA data. This was only the case for some days and was never more than a few minutes of data per day and never more that two consecutive minutes. This is not considered a problem, since 15 minutes intervals are used. When there are missing minutes in the data, this is interpolated, on the same way MoniBas does this (as described earlier). The maximum gap that is considered acceptable is 5 minutes, which is the same maximum type of missing data that MoniBas uses.

4.5 Information on the road sections

In order to give good information on the road sections these are divided for the different parts of the day. In this way the morning and the evening peak can be distinguished from other parts of the day. In table 4.3 the division is shown. The division between night and day is made, because at some roads there is a variable speed limit. The lower speed limit is during daytime. The times for day and night are based on the times that the variable speed limits are active.

Table 4.3: Definition interval times

		Start	End
Week day	Night	19:00	6:00
	Morning peak	6:00	9:00
	Day, off peak	9:00	16:00
	Evening peak	16:00	19:00
Weekend day	Night	19:00	6:00
	Day	6:00	19:00

For most roads there in information on vehicular length. The categories that are in the data are shown in table 4.4. For this research the division line is 5.6m. All vehicles shorter are regarded as short vehicles, all vehicles longer than 5.6m are regarded long vehicles is this research. It is chosen to make this simplification, so that a single value can be fed to the model, which is the percentage of long vehicles.

Table 4.4: Definition of vehicle classes

	Minimum length [m]	Maximum length [m]
Short vehicles	0.0	5.6
Medium vehicles	5.6	12.2
Long vehicles	12.2	—

All 15 minute instances are classified as either free flow or congested. The boundary for this is chosen at 70 km/h. In figure 4.4 a histogram is shown in which the prevalence of all speeds is categorized for the A27 near Lexmond. This histogram shows that the free flow conditions almost all have average speeds in the interval of between 90 and 120 km/h, while all congested intervals have average speeds of between 10 and 50 km/h. There are very little instances with an average speed between 50 and 90 km/h. Because of this the boundary for congestion and free flow is set in the middle at 70 km/h.

An overview the traffic performances on the roads is given in table 4.5. On the left for each road the percentage of the instances that are congested are shown, on the right the average speeds. These are shown for the total, for the morning peak (M.P.) and the evening peak (E.P). From this table can be

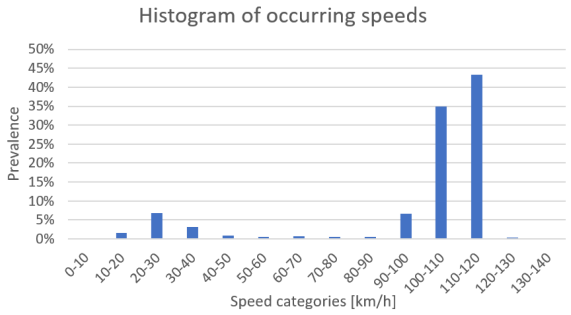


FIGURE 4.4: Histogram of occurring speeds at A27 near Lexmond.

seen that all roads perform very differently. Many road sections suffer from much congestion, some mostly during the morning peak, others mostly during the evening peak. On the A28 near Zwolle, a road section without a clear bottleneck, almost no congestion is registered. On the A7 there also is no clear bottleneck, but congestion occurs in the morning peak, this is probably caused by on ramps downstream of the measured location.

TABLE 4.5: Statistics of speed and congestion on the road sections.

road	% con.	% con. M.P.	% con. E.P.	av. sp.	av. sp. M.P.	av. sp. E.P.
A01_03	4.0%	33.8%	1.5%	96.0	61.5	111.6
A02_02	4.7%	43.3%	1.1%	95.4	63.4	107.0
A04_01	10.5%	72.2%	17.6%	79.0	43.7	80.2
A04_02	9.6%	17.4%	72.0%	80.3	81.6	41.6
A07_01	2.2%	20.2%	0.0%	104.3	83.2	113.2
A27_02	14.5%	11.5%	91.7%	65.9	85.9	25.2
A27_03	1.4%	1.1%	11.6%	97.3	97.2	82.3
A28_01	4.6%	33.1%	4.1%	96.5	68.3	100.3
A28_02	0.4%	3.0%	0.0%	111.6	105.5	114.4
A58_01	7.6%	8.7%	60.6%	83.9	94.3	45.5
A58_02	7.6%	14.8%	61.3%	82.8	81.7	42.0
A58_03	13.6%	30.6%	57.4%	75.0	66.2	48.2

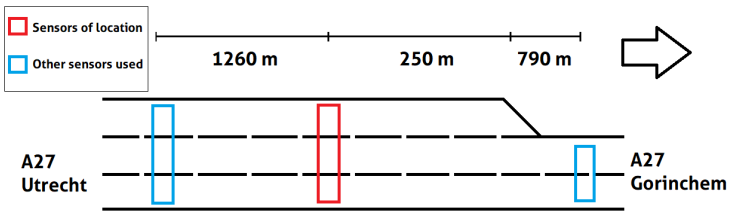
5 Testing several machine learning techniques

Different types of machine learning techniques were tested on their effectiveness. The place for this testing is the road section on the A27 near Lexmond. All testing has been done using WEKA. This model testing consists of two parts. In the first part an estimation of whether or not the traffic is in a congested state has been made. In the second part a speed estimation of the traffic has been made. For all techniques in which randomness is involved, the average of five different runs has been calculated, for some of those runs box plots have been made, so the range of the values can be identified.

5.1 Tested road section

For finding testing different kinds of machine learning models a stretch on the A27 near Lexmond is selected. At this point there is a lane drop from three to two lanes as can be seen in FIGURE 5.1B. This location is a bottleneck that gets activated almost daily. There are no other points of interest near this point that cause congestion. This means that (almost) all congestion occurs because of this bottleneck at this road stretch. In FIGURE 5.1A it can be seen that the detector that is researched is situated just before the bottleneck. Also the detector from upstream and the detector of just downstream the bottleneck are taken.

(A) Location



(B) Road configuration

FIGURE 5.1: Location and situation of the tested A27 road section near Lexmond.

There have been made statistics from this road section, that can be found in TABLE 5.1. There is almost no missing data on this data, only during one day there are two hours of data missing. This missing data is removed from the dataset. Because of the completeness of the data, recurrent models can more easily be applied. From these statistics can be seen that there is a lot of congestion on this road. During the evening peak there is almost always congestion, even in the weekends or during the night there sometimes is congestion. In the previous chapter is shown that this road has most congested intervals of all researched roads, which makes this road suitable for this research.

TABLE 5.1: Road statistics of A27

		Week day				Weekend day	
		night	morning peak	day off-peak	evening peak	night	day
Missing	[%]	0.0	0.0	1.3	0.0	0.0	0.1
Flow	[veh/h]	817.2	2725.7	2733.9	2515.9	912.7	2061.1
Speed	[km/h]	102.8	97.9	91.3	44.8	112.2	110.7
Con. ints	[%]	1.4	4.4	10.2	88.3	1.1	2.1

In FIGURE 5.2 a plot of flow and speed for a typical day have are displayed. Also taking into account all other days, it can be seen that the maximum intensity on this road lies around 4000 veh/h. When that number is reached, almost always congestion will occur. But also at much lower intensities often congestion occurs. It is very probable that hysteresis that Treiber and Kesting (2013) describe can be one of the causes. When a congested state is reached it is difficult for the system to restore to its full capacity. Before the system can reach full capacity again, the intensity has to drop first, so the system can come out of the hysteric situation. In the example shown, both in the morning– as in the evening peak congestion occurs. The evening peak is almost always present during weekdays. In extreme cases it already starts at 15:00h, and it can last up to 20:00h.

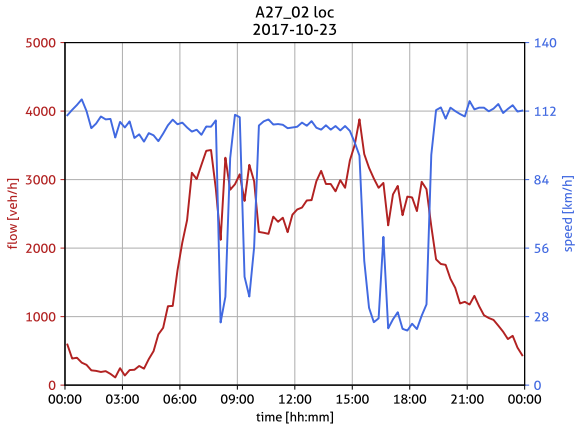


FIGURE 5.2: The situation on a typical day on the A27

5.2 Congestion estimation

For the estimation of whether or not congestion occurs three different kind of models are evaluated. These include a logistic regression model, decision tree learning models and neural network models. All models are evaluated by the f-score (as explained in EQUATION 3.2). The property that the f-score includes both the precision and recall, makes that this is a proper measure for comparing the different speed estimation methods, since it both includes instances that are classified as free flow, while they were in fact congested and it includes instances that are classified as congested, which were in fact not congested. All f-scores for the different models are summarized in TABLE 5.2.

5.2.1 Logistic model

The logistic model is in fact an ordinary regression model that can be used to make classifications. An advantage of this model is that it is relatively simple and very fast in execution. The output of the model is a number between 0 and 1 for each class, which represents the probability that the model gives to each classifications. All those probabilities add up to 1, the highest probability that is given is the estimation that the model makes. In this case with only two classes (congestion and free flow), the classification 'congestion' is made if the probability for congestion is higher than 0.5. The f-score for a normal logistic model is **0.61**.

From this logistic model a recurrent version is applied. This means that the estimated probability of congestion is fed back to the model and iteratively new estimations are made. The probability values that are fed back are the estimation for congestion of the estimated instance and of the instance just before it and just after it. So a total of three congestion estimations are fed back into the model. This process is visualized in FIGURE 5.3. In each step these estimations can be used to further improve the model. Having a recurrent model makes it possible for the model to look into the future and in into the past, which is of great help in this estimation, since what happens in the near past and near future can give much information on the current traffic state.

Several iterations have been made in this model, in which in every step the new congestion estimations

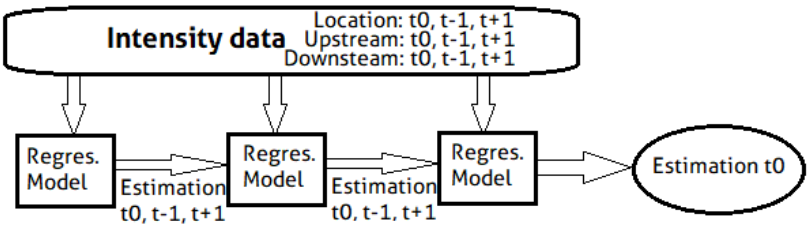


FIGURE 5.3: A schematic recurrent regression model with two iterations.

were used. This is shown in FIGURE 5.4. It is remarkable that the f-score drops after six iterations, after having increased very vast in the first stage. This might be the case, because the fed back information plays a very important role in the estimation. When one of the estimations is incorrect, it can have a big effect on further iterations. The optimum in this case lies around 6 iterations. Recurrent logistic regression can get the f-score up from 0.61 to **0.84**, which is an improvement of 38% from the non-recurrent logistic regression.

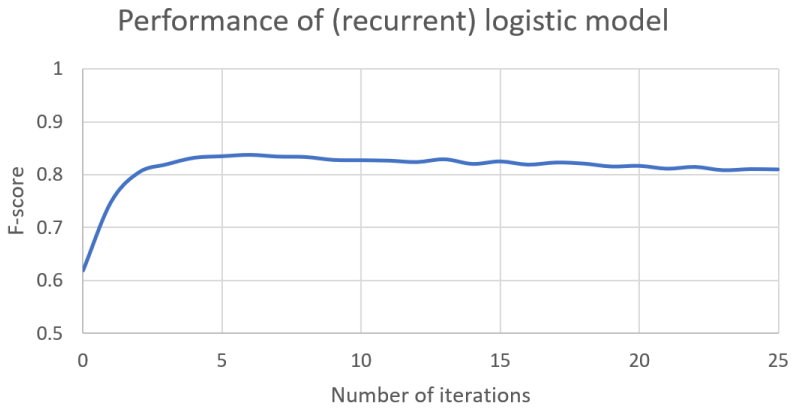


FIGURE 5.4: F-scores for the logistic model

5.2.2 Decision tree learning models

There are several DTL-algorithms included in WEKA, of which two have been tested. The tested J48 algorithm is an open source java implementation of the C4.5 DTL algorithm. This algorithm is a proven basic form of DTL, although it is possible to make some alterations in the algorithm. The most important options that can be chosen is the amount of pruning and the minimum instances a leaf must have. Those options have been researched for their performance on the estimation of congestion.

It was found that pruning the model did not improve it, but resulted in lower f-scores. Changing the minimum number of instances per leaf however did result in a better performance. When a higher minimum number of instances per leaf is set, the model becomes smaller, because branches with few instances are removed. Setting such a number may prevent overfitting, however when the number is too high, proper separations are not made.

Several values for the minimum number of instances per leaves have been tested. In FIGURE 5.5 the results are shown. The figure clearly shows that low and higher values give bad results. The best results in this case were found for values between 13 and 25. The best f-score of **0.60** was found with a minimum of 16 instances per leaf.

The J48 DT has been improved by making the model recurrent. The estimation that has been made for the instance on moment t have been fed back to the model, together with the estimation for moment $t-1$ and $t+1$. FIGURE 5.6 shows that this does improve the f-score from 0.60 to **0.72** at 6 iterations, after which no further improvement is found.

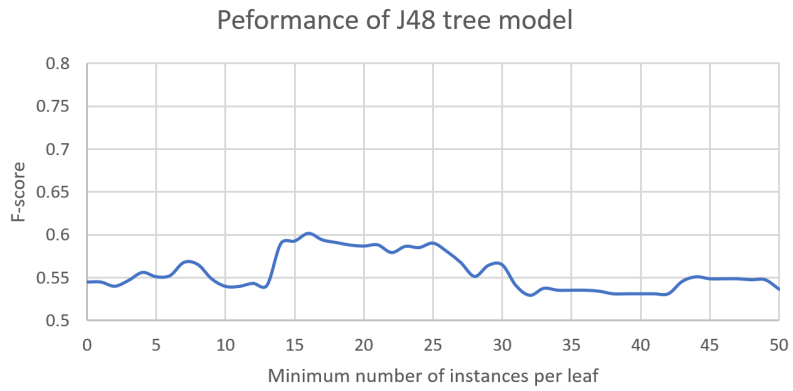


FIGURE 5.5: The performance of a unpruned J48 tree model. The minimum number of instances per leaf has been varied.

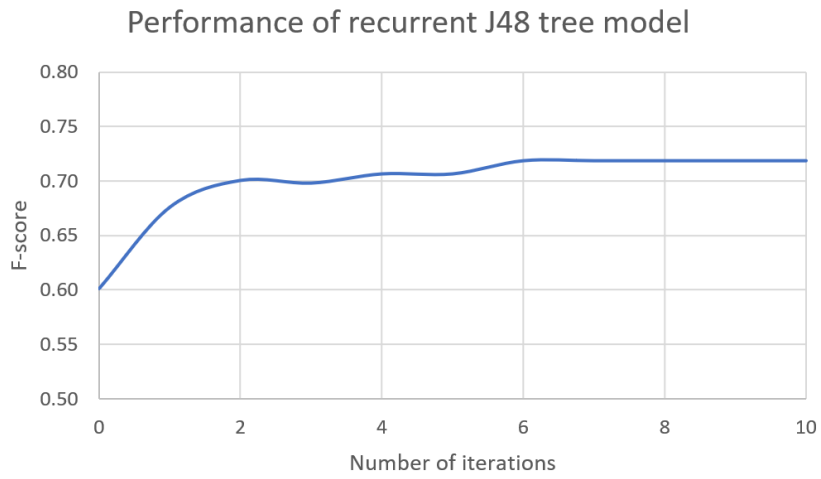


FIGURE 5.6: Based on a minimum of 16 instances per leaf, a recurrent model has been implemented.

A more complex implementation of DTL is the random forest algorithm. The random forest consists of many decision trees, which together make a forest. All those trees are built up independently from each other, so that different features can be combined in each tree. The classification that is made most by the trees wins and becomes the estimation for the instance. A box plot is made, because different seeds are used for generating the random forest.

In FIGURE 5.7 the box plot is shown with the results for a (recurrent) random forest. When comparing the non-recurrent result to the result of the J48 tree a better result is found with a f-score of 0.66. However when the recurrent random forest is applied, the performance does not improve as much in the J48 tree. After three iterations, no further improvement was found. The resulting f-score is 0.69 for the recurrent random forest, opposed to 0.72 from the recurrent J48 tree.

5.2.3 Artificial neural networks

The neural network that is used in this research is a network in which the inputs are the intensities (all represented by one node) and the outputs are two nodes, one for the probability of congestion and one for the probability of free flow. The standard neural network from WEKA is used. This neural network is a network in which all perceptrons are activated by the sigmoid function, all weights are initialized randomly and all inputs are normalized into values between -1 and 1. For a neural network there are some parameters that have to be set.

For the learning process the learning rate and the momentum are two important parameters, they decide the speed and the precision of the model. In FIGURE 5.8 the performance for the different configurations

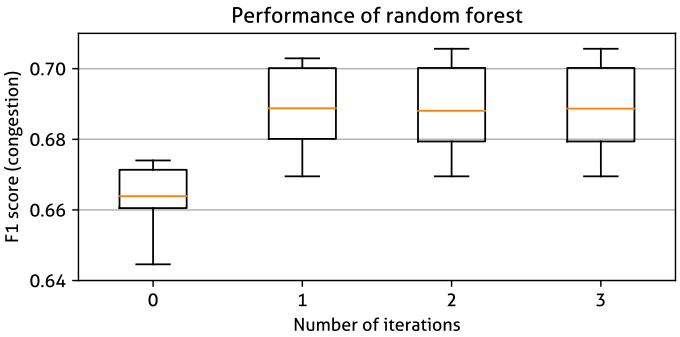


Figure 5.7: Performance of a random forest decision tree estimator. 0 iterations is the normal random forest estimator, no change was seen after three iterations in the model.

is given. It can be seen that a high momentum works well with a slower learning rate. In this case a combination of a learning rate of 0.05 and a momentum of 0.9 gives the best output. For finding these values, 500 epochs were evaluated. When a higher number of epochs was used, lower learning rates improved, but did not exceed the value found at the combination

F1-score ANN, 8 nodes in hidden layer

learning rate \ Momentum	0	0.5	0.9	0.95	0.99
0.1	0.77	0.81	0.81	0.81	nan
0.05	0.74	0.78	0.83	0.80	0.77
0.01	0.63	0.65	0.77	0.81	0.81
0.005	0.47	0.63	0.74	0.76	0.82
0.001	0.01	0.24	0.63	0.65	0.80

Figure 5.8: - scores for different combinations between the learning rate and the momentum on a neural network with 8 nodes in the hidden layers.

Another aspect of importance for the performance of a neural network is its size. This size is determined by the number of hidden nodes that are in the hidden layer(s) of the model. Results of applying different number of nodes in the model are shown in figures 5.9 and 5.10. Figure 5.9 give a box plot with the performance of the neural network, that consists of one hidden layer. The figure clearly shows that adding a lot of nodes does not improve performance. The highest value is found at 8 nodes in the hidden layer, with a f-score of 0.83.

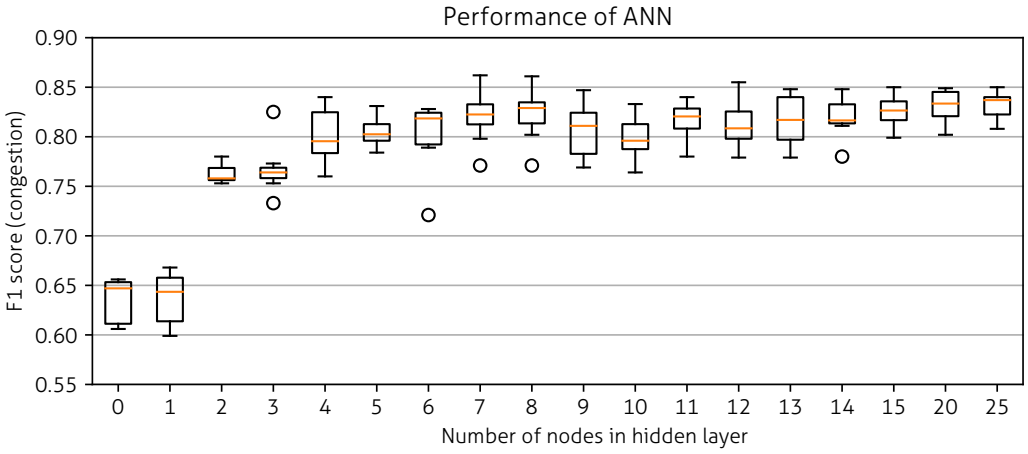


Figure 5.9: Performance of the neural network, based on the the number of nodes in the hidden layer.

Also the ANN was tested with two hidden layers, with different combinations of the sizes of both layers. Figure 5.10 shows that this gives good results, but no better results than having a single hidden layer ANN. The f-scores for both are no higher than 0.83.

F1 score ANN

layer 1 \ layer 2	4	8	12	16
4	0.79	0.81	0.80	0.81
8	0.82	0.82	0.83	0.82
12	0.83	0.81	0.83	0.83
16	0.82	0.83	0.83	0.83

FIGURE 5.10: Performance of a neural network with two hidden layers, there is no improvement on a single layer neural network.

The last ANN variant that is tested is the recurrent neural network (RNN). As in the previously discussed recurrent models, the RNN feeds its estimations back in the system, so it can look back in time and look forward in the future in every iteration. The RNN was tested on single hidden layer ANNs, with different sizes of the hidden layer, as can be seen in FIGURE 5.11.

The performance of the RNN is much better than that of the normal ANN. Already after one iteration the f-score increases substantially. After about 3 iterations a 10 node hidden layer RNN has a f-score of 0.90, which no further increases at more iterations.

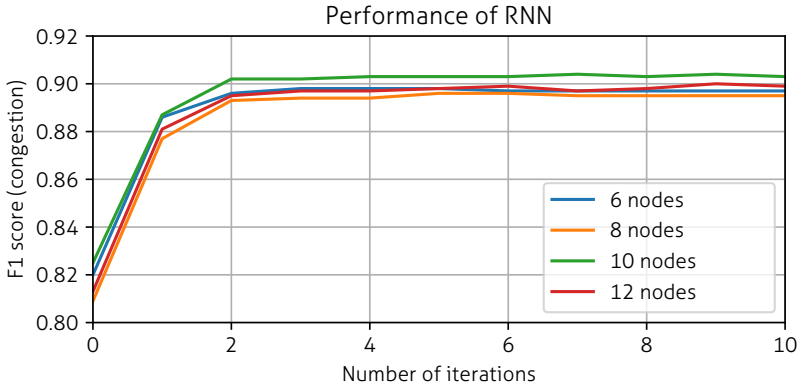


FIGURE 5.11: Performance of a recurrent neural network with a different number of nodes in the hidden layer.

5.2.4 Comparison

In TABLE 5.2 all f-scores have been summarized per category. The recurrent neural network gives the highest average score of **0.90**. The neural networks score better than the logistic– and decision tree models. It is remarkable that the logistic models score better than the decision tree learning models, while the DTL model is a supervised machine learning technique and the logistic model is an ordinary mathematical regression model.

In all cases the recurrent version of the model scores better than the non-recurrent version. This is not unexpected since traffic patterns are temporal sequences of the traffic state variables and in recurrent models those temporal sequences can be captured. These results do show that this indeed leads to a better estimation model.

In order to see what the congestion estimation looks like in the different models, three plots for a typical day have been made. These can be seen in FIGURES 5.12 to 5.14. The differences can be seen quite clearly. In this case the RNN estimation is flawless and estimates all congestion with high certainty correctly. The logistic– and DTL model however have more difficulties. They both miss out an entire (morning) congestion. Also the DTL model is very uncertain during the afternoon congestion.

5.2.5 Alternating the boundary of acceptance

An alteration to the boundary of acceptance has been tested on the RNN model. The RNN models has been chosen for researching this boundary, because it was the best performing model. In training and

Table 5.2: Performance of the different machine learning techniques.

Technique	F-score	Configuration
Logistic	0.61	–
Rec. Logistic	0.84	6 iterations
J48	0.60	unpruned, min. 16 inst. per leaf
Rec. J48	0.72	6 iters., unpruned, min. 16 inst. per leaf
Random Forest	0.66	–
Rec. Rand. For.	0.69	3 iterations
ANN	0.83	l.r.: 0.05, mom.: 0.9, 500 epochs, 8 hidden nodes
ANN (2 lay.)	0.83	l.r.: 0.05, mom.: 0.9, 500 epochs, 12, 4 hidden nodes
RNN	0.90	l.r.: 0.05, mom.: 0.9, 500 epochs, 10 hid. nod., 3 iter.

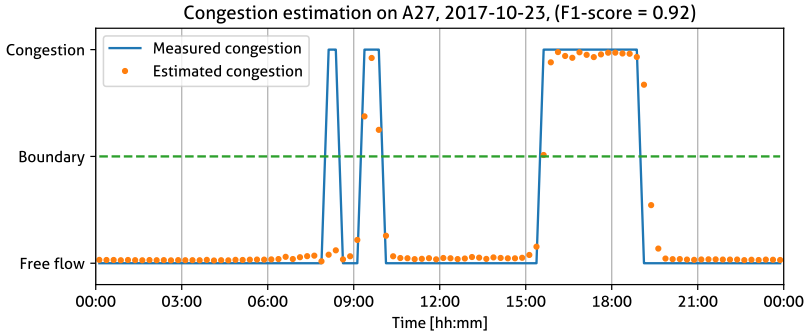


Figure 5.12: Congestion state estimation of recurrent logistic model on a typical day.

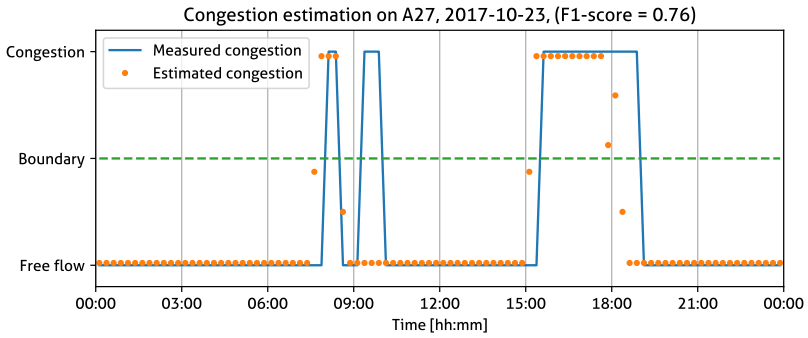


Figure 5.13: Congestion state estimation of recurrent J48 tree learning model on a typical day.

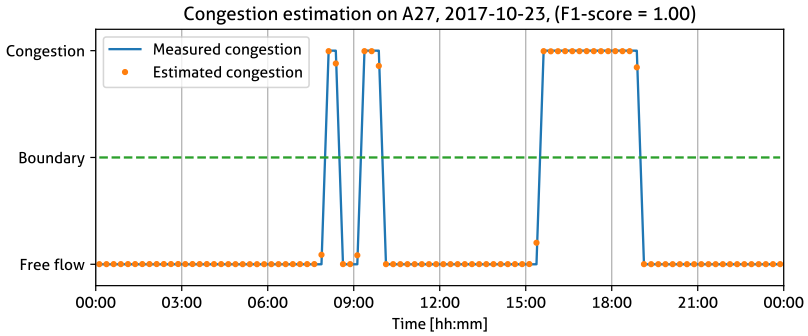


Figure 5.14: Congestion state estimation of recurrent neural network on a typical day.

testing the model an instance is considered congested when the congestion exit node has a value higher than 0.5. Although within the training process the model is optimized on this boundary value, results could theoretically become better by altering this boundary value.

This boundary has been alternated, with values between 0.01 and 0.99. The results of this alternation are shown in FIGURE 5.15. Logically it can be seen that the precision increases with a higher boundary, while the recall decreases. This can be explained by the definition of both precision and recall. The factor of error in recall is the false positive and with recall it is the false negative. With a boundary approaching 1 there will no longer be false positives, since hardly any instances are classified as positive. At the same time at high acceptance boundary there will be many false negatives, thus the recall decreases.

The f-score combines precision and recall and can therefore be considered as a compromise performance measure between those two. FIGURE 5.15 shows a very flat f-score when plotted to a variable boundary value. It is lower at very high and very low boundaries, but roughly between 0.1 and 0.9 it is almost constant. An explanation can be found in FIGURE 5.14, where can be seen that in the estimation of congestion in this model, the model is very certain of itself, only outputting very low and very high values. There are hardly any values between 0.1 and 0.9, so varying between this values does not change much. More extreme boundary values lead to lower f-scores, therefore there is no reason for changing the boundary value of 0.5.

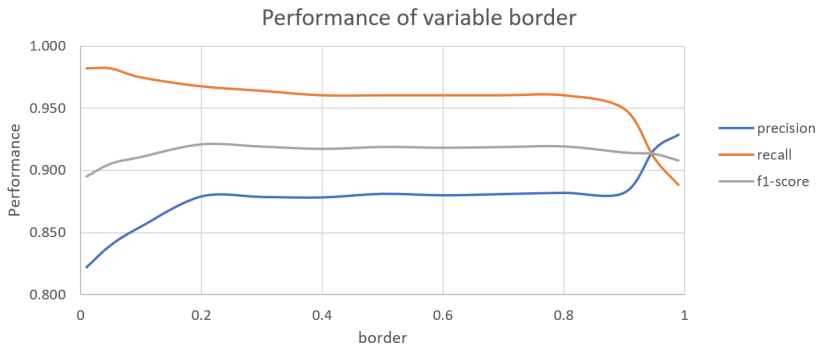


FIGURE 5.15: Performance with variable boundary.

5.3 Speed estimation

For speed estimation the speed is estimated based on intensities. This is a slightly different process that the congestion state estimation, since in that case there were only two categories, while with speed estimation a numeric value is estimated. The performance of these models are compared by the root mean squared error (RMSE, EQUATION 3.3), which is a value for how much the estimation is in line with the true value. Within this method big differences in real value and estimation have a big effect on the outcome of the performance measure.

5.3.1 Linear regression model

Also in speed estimation the first tested model is a regression model. The difference between the logistic version in congestion estimation is that in the logistic model the exit node generates a value between 0 and 1, indicating the probability of congestion, while in speed estimation the exit node gives a real speed. Actually this model is less complex than the logistic model, since the categoric scaling part of the logistic model can be omitted.

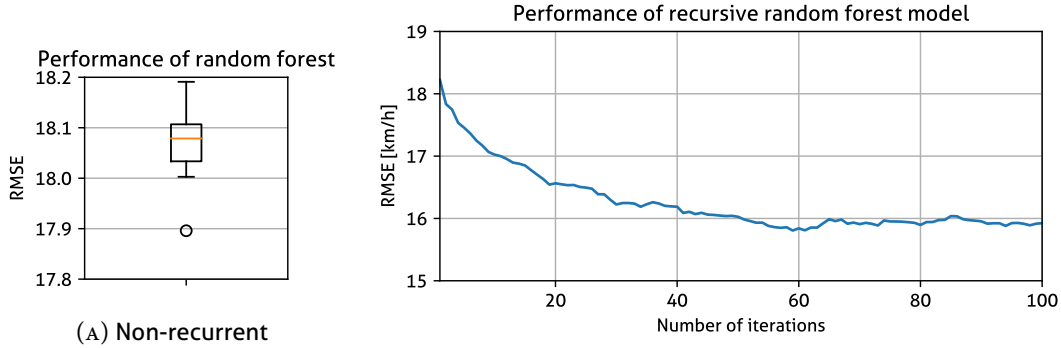
The output of a non-recurrent regression model was an RMSE of 26.2 km/h. This improved to a RMSE of 23.7 km/h after 13 iterations in the recurrent version, after which the performance is constant (FIGURE 5.16).

5.3.2 Decision tree learning models

For DTL only the random forest is applied, since the J48 decision tree can only be applied when data is categorized into a discrete number of categories. In FIGURE 5.17 the performance of the random forest model is shown. In figure A a boxplot of a non-recurrent version is shown, in figure B the recurrent version. It is remarkable that it takes many iterations before the performance improves. With a RMSE

Figure 5.16: Performance of linear regression model. The RMSE decreases until 13 iterations, after which it is stabilized.

of 18.1 km/h for the non-recurrent version and 15.8 km/h for the recurrent version the random tree performs significantly better (13%) than the regression model.



(a) Non-recurrent

(b) Recurrent

Figure 5.17: Performance of the (recurrent) random forest model.

5.3.3 Artificial neural networks

For speed estimating neural networks of different sizes are tested. Except for its size and its output node, the specifications of the network are the same as in congestion estimation. The neural network is capable of directly estimating speeds. In figure 5.18 a box plot is shown with the results for an ordinary non-recurrent ANN. An RMSE of about 20 km/h was found, with not much difference between the different model sizes.

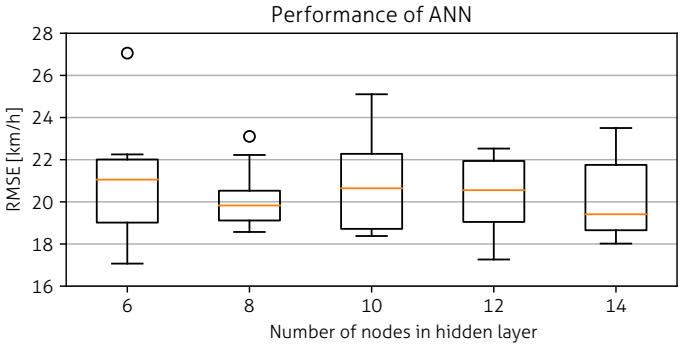
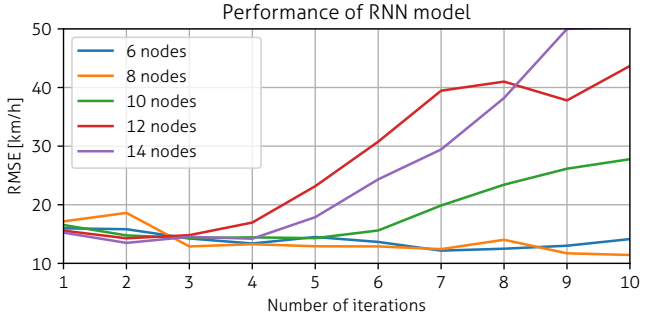


Figure 5.18: Performance of neural network, with several values for the number of nodes per hidden layer.

When testing the RNN for speed estimation, a difference was found between the single layer and the

multilayer neural network. FIGURE 5.19A shows that some models start to diverge at a higher number of iterations. With about three iterations, improvement is shown, but after these iterations those models (with a higher number of nodes in the hidden layer) start to perform very badly, resulting in high RMSEs. The reason for this is unclear. A possible explanation is that the information that is fed back to the system is a big factor in the estimation in the iteration step. When there are some small errors in the information that is fed back (there will always be errors in this info), each iteration can build on on these errors, and it can start to get worse at every iteration. Even though this is the most likely explanation, that does not explain why it happens in some model configurations, while it does not happen in others. For example it does not happen in the model with 8 and 10 nodes in the hidden layer, nor did it happen in the multilayer configuration.

Those multilayer configuration of the RNN gives the best results on speed estimation. In FIGURE 5.19B results of this are shown. For these results five iterations in the model were made. After five iterations there was no improvement in the scores, but the previously discussed divergence after many iterations did not happen either. This RNN delivers scores for the RMSE of under 10 km/h. This means that on average the estimated speed is fairly close to the real, measured speed. This difference is at least small enough to give a good indication of the speed. An example in FIGURE 5.22 is discussed in the comparison between the models.



(A) One hidden layer

RMSE of RNN with two layer configuration

layer 1 \ layer 2	4	6	8	10	12
4	10.4	9.8	10.6	10.3	10.3
6	10.4	9.9	9.7	10.3	9.4
8	10.9	10.5	9.6	9.8	9.7
10	10.5	9.6	9.7	9.6	9.4
12	10.2	9.8	9.5	9.8	9.5

(B) Two hidden layers

FIGURE 5.19: Performance of the (recurrent) random forest model.

5.3.4 Comparison

A comparison has been made for all models tested and is summarized in TABLE 5.3. The regression models perform worst of all models tested. Even the recurrent version of this scores much worse than all other models, even though it improves a little on the non-recurrent version. When comparing the non-recurrent versions of the random forest DT and the ANN, it was found they have similar scores, the random forest even scores a little better, which is remarkable, since in congestion estimation the ANN scored much better than the decision tree learning model.

The best scoring models again were the recurrent versions. When the recurrent version of the neural network is compared to that of the decision tree learning model, a bigger improvement is seen by the neural network. This is the best scoring model in this case, with a RMSE of 9.4 under the best configuration.

In FIGURES 5.20 to 5.22 an illustration is given of these results on a day with congestion. Also scatter plots have been made plotting all measured and estimated speeds of the test set. The linear regression line has been plotted on it and the r^2 value is displayed. The function for the linear regression line should optimally be $y = x$.

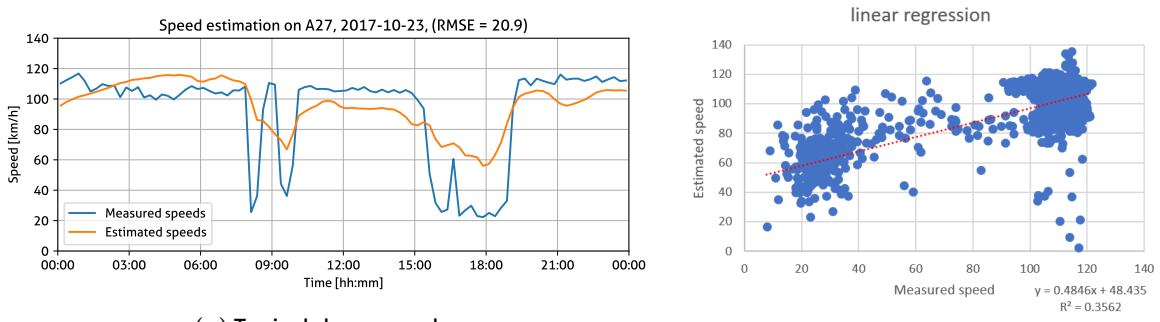
When viewing the regression model it can be seen that this does not give a good impression of the speed. The estimated speed is a little lower during congestion, but it does not really follow the measured speeds. This example shows that linear regression is unsuited for speed estimation. In the scatter plot it can be easily seen that the function does not suit well. The estimated speeds are too high, indicating

Table 5.3: Comparison in speed estimation

Technique	RMSE	Configuration
Linear regression	26.2	–
Rec. lin. regr.	23.7	13 iterations
Random Forest	18.1	–
Rec. Rand. For.	15.8	59 iterations
ANN	19.4	l.r.: 0.05, mom.: 0.9, 500 epochs, 14 hidden nodes
RNN	11.4	l.r.: 0.05, mom.: 0.9, 500 epochs, 8 hid. nod., 10 iter.
RNN (2 lay.)	9.4	l.r.: 0.05, mom.: 0.9, 500 epochs, 10, 12 hidden nodes, 5 iter.

that congestion often remains undetected, but also the speed for congested circumstances is estimated far too high.

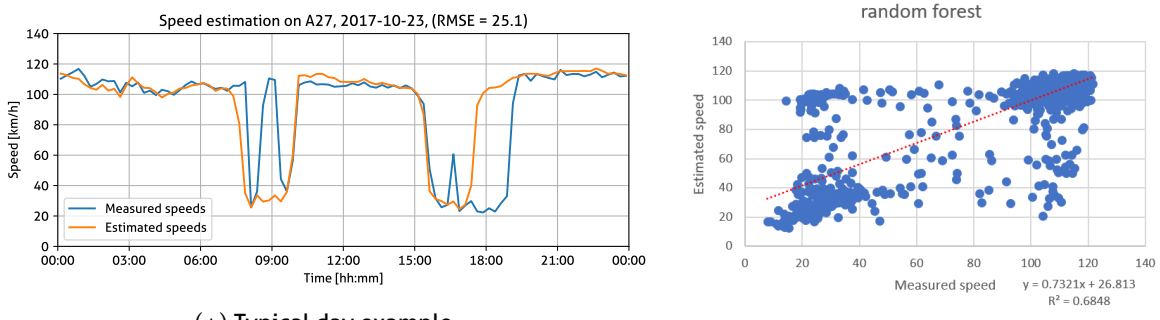
The recurrent random forest model and the RNN give a much better estimation of the speeds. Both these models follow in the example day the actual speeds quite well, the RNN better than the random forest. In the scatter plots it can be seen that especially the RNN performs well. A value of $r^2 = 0.89$ was found here, while the regression line is very close to $y = x$. Also the RNN makes some mistakes (both of type I as type II), but the majority of all instances are estimated close to the measured speed.



(A) Typical day example.

(B) Scatter plot.

Figure 5.20: Performance of recurrent linear regression model.



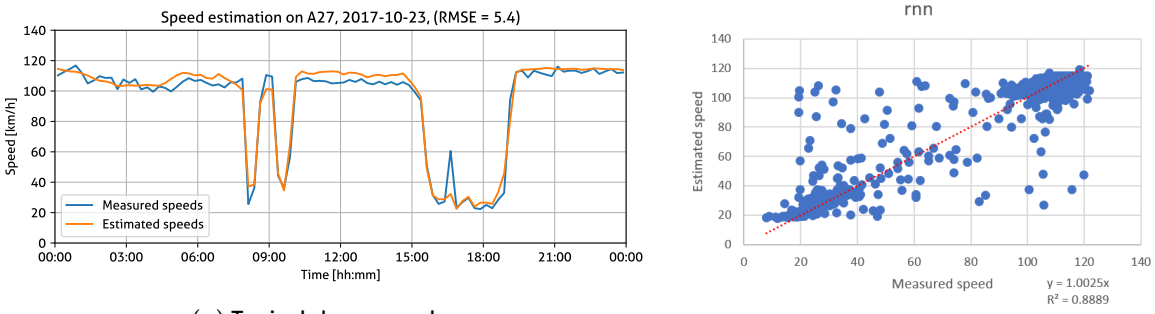
(A) Typical day example.

(B) Scatter plot.

Figure 5.21: Performance of random forest model.

5.4 Validation on other roads

All testing took place at one road section on the A27. This testing is validated on the other road sections in this research. For this validation the RNN with the best scoring setting was chosen. In this validation all roads data sets are split in training and test sets, so every road is tested using a model that is also trained on that specific road. The training set and test set are separated by the same dates as the A27 was separated, meaning two third of the instances were used for training and one third of the instances is used for testing. This means a different model is made for each road section.



(A) Typical day example.

(B) Scatter plot.

FIGURE 5.22: Performance of RNN model.

In TABLE 5.4 the results of the validiation are shown and the results differ a lot from each other. These differences have to do with the characteristics of the roads, that were discussed earlier and can be seen in TABLES 4.1 and 4.5. On roads on which the bottleneck is a lane drop congestion estimation leads to high f-scores.

TABLE 5.4: Results of validation on other road sections.

road	f-score	std	RMSE	std
A01_03	0.715	0.041	11.16	0.30
A02_02	0.664	0.010	9.65	0.65
A04_01	0.769	0.011	10.45	0.78
A04_02	0.861	0.026	8.77	0.87
A07_01	0.621	0.029	9.42	0.85
A27_02	0.901	0.013	9.44	1.00
A27_03	0.391	0.028	13.47	0.25
A28_01	0.479	0.026	15.10	0.89
A28_02	0.578	0.031	6.84	0.17
A58_01	0.805	0.021	10.37	0.55
A58_02	0.780	0.032	11.49	1.02
A58_03	0.870	0.011	9.99	0.45

Very low f-scores were found on the A27 near an on ramp and narrow lanes and on the A28 near a ramp. An explanation for the lower scores in general at places with a ramp is the fact that not all traffic intensities are there in the model. For example when there is a off-ramp between the upstream location and the measured location, there is no conservation of flow, since some of the flow will have taken the off-ramp.

The two locations without a particular bottleneck have f-scores of around 0.6, scoring better than the locations with a ramp, but worse than the locations with a lane drop. A cause of the relatively low scores is that on places without a particular bottleneck there is not much congestion, so these places have relatively few instances of congestion, which makes training harder. When comparing the RSME of these sections without bottleneck, they score relatively good, when comparing to their f-score. This can be explained by the fact that there are few instances of congestion, so estimating a speed near the 'normal' speed will deliver good results.

Taking all these things together, it can be said that the RNN model performs well on sections with a clear bottleneck in which all traffic intensities are included in the model. If intensities are missing, due to ramps, this makes results become worse. In cases without a bottleneck the speed estimation performs well, while congestion estimation is harder.

6 Additional input attributes

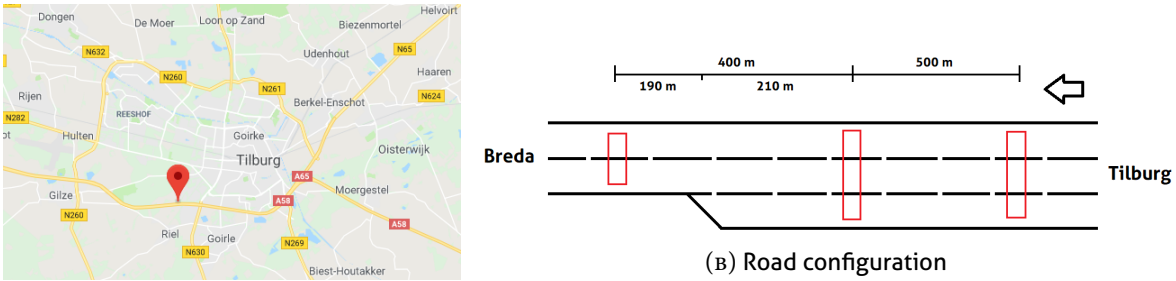
In this chapter the influence of additional attributes is tested. In the previous chapter different machine learning methods were discussed on a specific road section, only feeding intensities to the model. In this chapter three different types of extra data are included, which are the weather conditions, the percentage of long vehicles, and the variance of the intensities within the interval. The effects of these extra factors are tested and evaluated.

6.1 Tested road section

For testing additional factors a road segment on the A58 has been chosen. Previous (model) testing took part at a road section at the A27, which was very suited for that purpose. This road section however did have one disadvantage, it has no information on vehicle length available. Since the A58 has this information, this road is chosen for testing the influence of additional input attributes.

6.1.1 Description of road section

The road section of the A58 is located in the west of Tilburg, just downstream of on-ramp Goirle, as is shown in FIGURE 6.1A. The point that is measured is in western direction, out of the city of Tilburg. Because of this direction, intensities are highest during the afternoon peak hour. There is a bottleneck located at this road section, which is a lane drop from three to two lanes FIGURE 6.1B. This bottleneck often causes congestion. The researched location locates just upstream from the lane drop.



(A) Location

(B) Road configuration

FIGURE 6.1: Location and situation of the tested A58 road section near Tilburg.

An overview of statistics has been made in TABLE 6.1, to illustrate the traffic patterns of the road section. High flows occur both during morning peaks and evening peaks, the highest numbers during evening peak, as is expected, since it is an outbound motorway from a larger city. Because of this high intensities during evening peak, most congestion occurs during that time of day. Over 60% of all measured intervals during evening peak (4 – 7 PM) are congested, having an average speed during that period of only 42 km/h. In FIGURE 6.2 a typical day is shown, in which the high intensities during peak hours and the congestion during evening peak can easily be identified.

The share of long vehicles varies over different parts of the day. During peak hours the share of long vehicles is lower, while at night it is higher. There is a chance these patterns could lead to a certain bias in the machine learning model. Instead of combining the data on long vehicles, it could just associate high shares of long vehicles with nightly conditions and automatically estimate free flow conditions. When this attribute would become too important in the model, it could potentially estimate all instances with a high number of long vehicles as free flow, also in daytime instances. The chance that this bias is problematic is rather low, because together with the long vehicle share the intensities are still included as attributes. This probably already gives an indication of the time of day. Therefore it is not expected that adding long vehicle data would give more bias than only feeding intensities would give.

TABLE 6.1: Road statistics of A58 near Tilburg

		Week day				Weekend day	
		night	morning peak	day off-peak	evening peak	night	day
Missing	[%]	0.0	0.0	0.0	0.0	0.0	0.0
Flow	[veh/h]	729.2	2865.8	2608.9	3029.2	789.5	1918.9
Speed	[km/h]	112.6	81.7	98.6	42.0	118.0	116.0
Con. ints	[%]	0.1	14.8	3.7	61.3	0.0	0.3
Long veh.	[%]	21.2	11.9	16.1	5.6	6.7	6.0

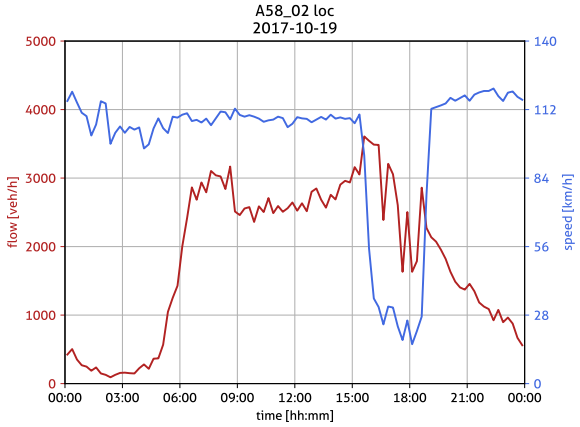


FIGURE 6.2: Flow and speed on a typical Thursday in October.

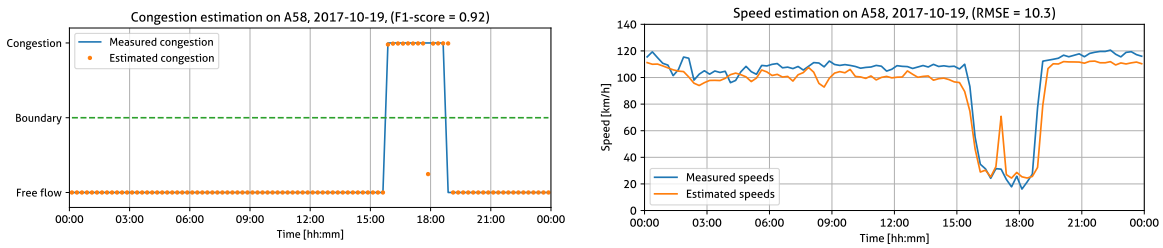
6.1.2 ML performance on road

Before testing additional attributes a base estimation has been made. The machine learning model that is used for testing is the same that gave the best results in the comparison of machine learning methods. For congestion estimation this was an RNN with 10 nodes in the hidden layer and doing three iterations in time. For speed estimation the best method also was the RNN, but with 5 iterations in time and two hidden layers of 10 and 12 nodes respectively. The comparison of all outcomes is the f-score for congestion estimation and RMSE for speed estimation.

First an estimation of congestion state has been made for the A58. These results already give relatively good results, showing that many situations are correctly classified. The number of cases in which congestion remains unidentified is much higher however than the number of cases in which congestion is falsely estimated. In total five runs with different seeds in the model were made. This led to an average f-score of **0.780** with a standard deviation of 0.032. In FIGURE 6.3 the estimation for congestion on a typical day is shown.

Also a base on speed estimation has been made out of five different runs. An average RMSE of **11.48** was found with a standard deviation of 1.02. In FIGURE 6.3 the estimation of speed is shown on a specific day. In speed estimation on that day can be seen that the estimated speed closely follows the measured speed, there is only one little difference in the evening peak. The same mistakes seems to be made in congestion estimation where one time during the evening peak the model suddenly estimates free flow, while in a congestion. But on average these models capture the traffic patterns very well.

Compared to the results of the A27 in the previous chapter, the model performs slightly worse, as already could be seen in TABLE 4.5. An explanation for this can be that on the A27 there were even more cases of congestion than on the A58, giving the model more opportunity to be trained to recognize congestion. Still, the results on the A58 are good for testing additional attributes, as it already gives quite good results, but there is still a place for improvement.



(A) Congestion estimation

(B) Speed estimation

Figure 6.3: Congestion and speed estimation on the A58 on a typical thurday.

6.2 Tested additional data

The results of the additional attributes that are tested are displayed. In figures 6.6 and 6.7 a comparison of all tested attributes can be found. All results are discussed in the comparison.

6.2.1 Influence of weather

Studies have shown weather influences traffic performance. Goodwin (2002) gives an overview on types of weather that influence traffic performance. Rain, snow, hail, fog and other conditions have an influence on traffic speed and capacity (Goodwin, 2002). For example rain was found to cause 10% reduction of speed. Because of the possible effects weather on speed estimation, it was chosen to research the effects of these conditions on the model.

Weather data from the KNMI weather station of Gilze-Rijen is used as a source. This weather station is location approximately 8 kilometres from the road section. It is assumed that weather conditions on the weather station are the same as on the road section. KNMI provides hourly data on whether or not one of the following conditions occurred: Fog, rain, snow, thunder and ice. Since snow and ice did not occur at all in the data and thunder only in less than 1% of the instances (all instances were only in the training set), it was chosen not to include these factors. Rainy conditions were there in 25.1% of the instances and foggy conditions were there in 3.6% of the instances.

On congestion estimation the average f-score is **0.777** with a standard deviation of 0.019. When estimating speed an average RMSE of **11.63** was found with a standard deviation of 0.74. Both of these results do not significantly differ from the base model, therefore it is concluded in this case that information on rainy and foggy conditions does not improve the model, so it is not useful to keep including it.

6.2.2 Different vehicle categories

For many road sections there is information on the share of different vehicle length categories. It is expected that information on this does influence model outcome, especially on speed estimation. Long vehicles generally have a lower speed limit than ordinary cars (80 km/h for trucks, 90 km/h for cars with trailer and 80 or 100 km/h for buses). For example in periods with low traffic intensity it is expected that the average speed is lower when there is a higher share of long vehicles, because the average speed limit for each vehicle is lower in that situation.

In order to test whether extra information on vehicle length improves the model the percentage of vehicles with a length of more than 5.6 meters has been given as an extra input. This size is provided in most of the NDW data and makes the difference between ordinary person cars and longer vehicles. Adding this vehicle length attribute as an input resulted into better results in both congestion estimation as in speed estimation. In congestion estimation this led to a f-score of **0.841** with a standard deviation of 0.016. In speed estimation the RMSE was **8.85** with a standard deviation of 0.67.

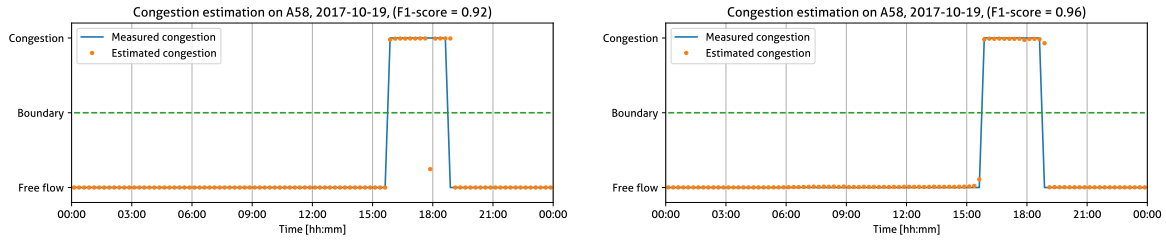
6.2.3 Deviation in intensity data

The data that is given as an input to the model is aggregated to 15 minutes, while the source NDW data that is used comes at a one minute resolution. This aggregated data only has the average intensities over the interval that is constructed of 15 separate values. As has been described in the methodology it is intentionally chosen to work with 15 minute intervals. Within this choice however it is possible to use more information from the original data. The additional data which is tested on its influence on the model is the deviation of the 15 intensity values. In stationary conditions (especially free flow situations, but also in congested situations) the deviation in the intensity data is low, while in situations where the deviation in intensities is higher, this likely indicates a change in the situation. A higher deviation for example is expected at the moment a congestion is formed or dissolved. Because of this property adding the deviation is expected to have a positive influence on the model. It is chosen to add the standard deviation of the intensities on the researched location.

It was found that adding the standard deviation to the intensities improves the estimations that are made by the model. In congestion estimation the f-score was **0.859**, with a standard deviation of 0.023, which is more than 10% higher than the base model. In speed estimation an RMSE of **9.82** with a standard deviation of 0.64 was found, which is a considerately better score than the base measurement.

6.2.4 Combined results

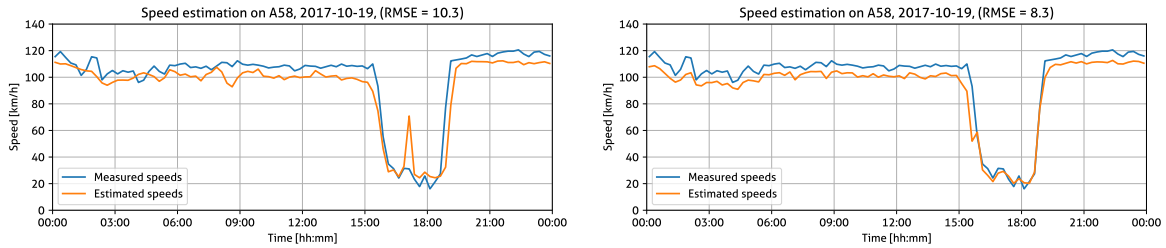
Of the three tested additional attributes to the model two have showed an improvement to the scores of the models. While weather conditions did not influence the outcome much and resulted into similar results, adding information about vehicle length and about the deviation in intensities resulted into higher scores. Because of these results these two factors have been added both to as input attributes and its effects were measured. In congestion estimation adding both these factors resulted into a f-score of **0.865** with a standard deviation of 0.017 and in speed estimation this resulted in a RMSE of **8.61** with a standard deviation of 0.94. In FIGURES 6.4 and 6.5 the results of estimation on a specific day are shown.



(A) Base congestion estimation

(B) Improved congestion estimation

FIGURE 6.4: Estimation on the A58, using vehicle length and intensity variation data, on a typical day.



(A) Base speed estimation

(B) Improved speed estimation

FIGURE 6.5: Estimation on the A58, using vehicle length and intensity variation data, on a typical day.

6.2.5 Comparison

In FIGURE 6.6 a comparison of f-scores is shown for all tested attributes on congestion estimation. In this box plot all of the previously discussed findings are summarized. This figure again shows that including weather in the data gives no improvement on estimation. Adding the share of long vehicles

and deviation within the interval does improve the results. A combination of these two gives the best results. The f-score increases from 0.78 to 0.87.

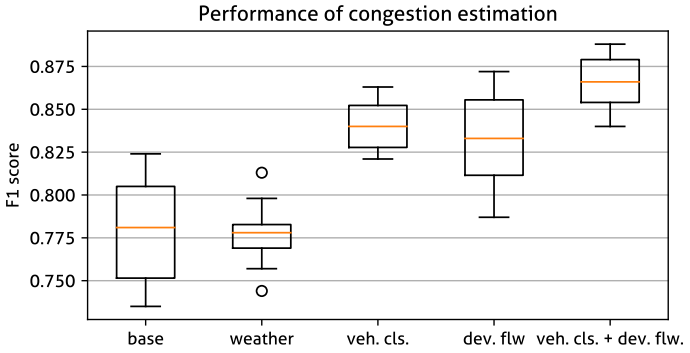


Figure 6.6: Comparison of f-scores with different additional attributes.

The same can be seen in speed estimation (figure 6.7). Also here a combination of adding the share of long vehicles and deviation gives the best results. The RMSE can be reduced using this additional data from over 11 km/h to 8.6 km/h.

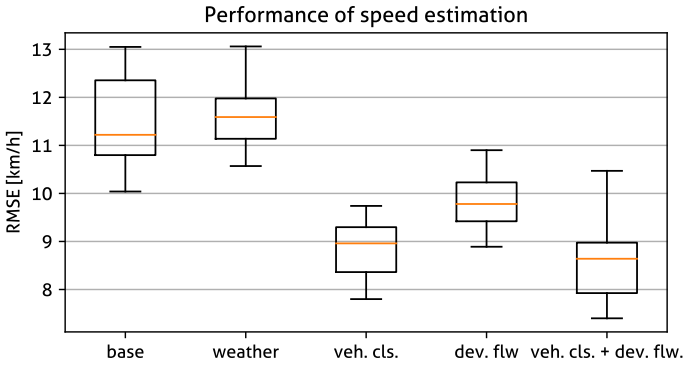


Figure 6.7: Comparison of RMSEs with different additional attributes.

6.3 Validation on other road sections

The results of this are tested on the other road sections is this research. The combination of additional attributes is added to the base dataset and tests have been run. For two road sections no information on vehicle length was available, this attribute is not included in those tests, so only deviation within the interval is added.

In table 6.2 the results for all road sections are shown. In congestion estimation on average a small improvement is found. Two locations show an extreme change in the f-score, these are both locations that scored bad in the original test. In speed estimation on average there is improvement to the situation without additional input attributes. Only the locations that have no information on vehicle length score worse.

TABLE 6.2: Improvement on road sections by adding attributes 'deviation in speed' en 'percentage long vehicles'. (* indicates a road section with no information on the length of vehicles.)

road	f-score base	f-score add.	improvement	RMSE base	RMSE add.	improvement
A01_03	0.715	0.689	-3.6%	11.16	11.27	-1.0%
A02_02	0.664	0.664	0.1%	9.65	9.24	4.2%
A04_01*	0.769	0.781	1.6%	10.45	11.81	-13.0%
A04_02	0.861	0.920	6.8%	8.77	6.82	22.2%
A07_01	0.621	0.641	3.3%	9.42	9.16	2.7%
A27_02*	0.901	0.892	-1.0%	9.44	10.50	-11.2%
A27_03	0.391	0.519	32.9%	13.47	13.14	2.4%
A28_01	0.479	0.500	4.5%	15.10	14.40	4.7%
A28_02	0.578	0.227	-60.7%	6.84	6.24	8.9%
A58_01	0.805	0.837	4.0%	10.37	9.69	6.5%
A58_02	0.783	0.865	10.5%	11.49	8.20	28.6%
A58_03	0.870	0.892	2.5%	9.99	9.18	8.1%

7 Multiple road input

Until this point in this research all model testing has solely been done on models that were trained with data from the same road. It is more interesting what the performance of the test sets will be that are trained on data that contains more than only the test road, or even on data that was not included in the train set at all. This chapter describes the performance of those kind of models.

7.1 Attribute selection

Since until this point all model testing has been carried out using the same road data as it was trained on, road characteristics did not have to be fed to the model. Because on the same road this data is all the same and would not add any extra information to the model. When multiple roads are included in the same dataset, these differences will be important, because by adding this extra information the model will 'know' the properties of the road section. The following attributes are added as input attributes:

- Speed limit [km/h]
- Number of lanes [-]
- Presence of on ramp on the road section [yes/no]
- Presence of off ramp on the road section [yes/no]
- Presence of a lane drop on the road section [yes/no]
- Distance to upstream location [m]
- Distance to downstream location [m]

All road sections do have their own capacity. This capacity is the maximum number of vehicles a road can process per hour. Because this differs per road, it is difficult to compare the meaning of values of intensities of different road sections with each other. To overcome this all intensities have been normalized to a percentage of its road's capacity. The capacity is determined by the boundary value at which 99% of all intensity values of the specific road are smaller than the number of capacity. By taking this value at 99%, outliers of intensities are not taken as the capacity value. After the capacity is determined all other intensity values are represented as a percentage of this capacity. This makes that the intensities of the different roads can be compared to each other, since they are both a comparable percentage.

7.2 All roads in both sets

The first test that has been carried out is a test in which the training data consists of the training sets of all researched roads combined. The size of the training set is about 48000 instances. Both a congestion estimation and a speed estimation has been made, using the most optimal models from the previous chapter. This is the RNN for both cases. Five runs were made for each test, with different random seeds and the average f-score is calculated.

This model is tested on the test sets of all road combined. For congestion estimation this resulted in an average f-score of **0.794** and for speed estimation in a RMSE of **11.86**. These scores are slightly worse than the testing in the previous chapter, but still provide a good model for estimating. These models can estimate congestion state or speed almost as good as the model that is only trained on the specific road is tested on.

Another test was done on the same training set (training sets of all roads combined). But the difference to the previous test is that the test set is not a combination of all test sets, but it is tested on all roads individually. Those scores are compared to the scores where the road is tested on a model that is only trained on the specific road. The results of this testing is shown in TABLE 7.1.

On average the scores on the test of the merged training set are slightly worse than the base score, which is determined in the previous chapter. The differences are small however. A remarkable result is the road A28_02 near Zwolle, where the congestion estimation improves, but the speed estimation became significantly worse. This can be explained by the fact that this is one of the roads without a bottleneck. By merging the training data, this road has more examples of other roads of what is a congestion, so the estimation of congestion improves. However this increased training set also causes confusion on the speed estimation as now there is much more variation of speeds in the training set.

The results show that it possible to make a merged set and still estimate congestion and speed. Even though on average the scores are lower, they are still of a quality level that estimations can be made with a high certainty.

TABLE 7.1: F-scores and RMSEs for the merged set.

road	f-score base	f-score merged	diff.	RMSE base	RMSE merged	diff.
A01_03	0.689	0.684	-0.7%	11.27	12.28	-9.0%
A02_02	0.664	0.723	8.9%	9.24	10.06	-8.9%
A04_01	0.781	0.760	-2.7%	11.81	11.24	4.8%
A04_02	0.920	0.931	1.2%	6.82	7.52	-10.3%
A07_01	0.641	0.560	-12.6%	9.16	11.20	-22.2%
A27_02	0.892	0.850	-4.7%	10.50	11.27	-7.3%
A27_03	0.519	0.483	-7.0%	13.14	13.29	-1.1%
A28_01	0.500	0.506	1.1%	14.40	14.80	-2.8%
A28_02	0.227	0.263	15.9%	6.24	8.10	-29.9%
A58_01	0.837	0.867	3.5%	9.69	11.29	-16.5%
A58_02	0.865	0.735	-15.1%	8.20	12.20	-48.8%
A58_03	0.892	0.856	-4.0%	9.18	9.82	-6.9%

7.3 Different roads in train set and test set

The last test that has been carried out, is one in which models are tested on data from roads it is not trained on. This is the most tricky test of all tests carried out, since testing takes place on a trained model, that does not contain examples from that specific road. This means that properties of other road sections must give enough information on patterns of congestion, that the model is able to transfer these properties to other road sections.

This model is expected to perform worse than other models tested because of the fact that all information must be transferred from other road sections. But since it is interesting to see whether or not this model still has some estimating power, this model is tested. It is possible that on roads that have similar properties as other roads in the set, relatively good results are found.

This testing is carried out by creating a training set that contains data of all roads, except for the road that is tested on. Also the test sets of the other roads are included, in order to create a bigger training set. This means that for each road section that is tested, a unique training set has been created. Testing was done using both the original training and test set of the road section that is tested.

Results of these tests are shown in TABLE 7.2. It is directly visible that the performance of testing on all roads has dropped a lot. Not a single road section shows performance that is comparable to that of tests that were taken before. Even though worse results were expected, these are very low scores, which cannot be used for estimating traffic conditions on roads.

In congestion estimation the f-scores have dropped almost all. Only a few road sections still have a f-score of over 0.5, which is a very low estimating power. The highest scores are found on the A27 near Lexmond, A58 near Tilburg and A58 near Breda. These are all locations with a lane drop from three to

two roads. It is possible that because of these similarities, the model could more or less successfully transfer properties from one road section to the other

In speed estimation there is also a big decrease in performance, but a variety is seen among the results. A few models now have vary bad scores for the RMSE of over 50 km/h, one road section even has a RMSE of over 80 km/h. With these scores the model has no power to say anything useful about speed on the road sections. The best scoring road sections have RMSEs between 10 and 20 km/h. This is the case for example on two road sections on the A58.

These results show that when one wants to estimate congestion or speed using a recurrent neural network, training data must include data from the same road section as is tested on. Having roads with similar properties in the training set helps to get better scores, but still these scores are very low and not really useful when an estimation of congestion and speed is required.

Table 7.2: F-scores an RMSEs for testing on a set that is not included in the training set.

road	f-score base	f-score not incl.	diff.	RMSE base	RMSE not incl.	diff.
A01_03	0.689	0.068	-90.1%	11.27	82.42	-631.3%
A02_02	0.664	0.501	-24.5%	9.24	13.83	-49.7%
A04_01	0.781	0.238	-69.5%	11.81	29.60	-150.6%
A04_02	0.920	0.415	-54.9%	6.82	51.51	-655.2%
A07_01	0.641	0.159	-75.2%	9.16	19.02	-107.7%
A27_02	0.892	0.660	-26.0%	10.50	39.55	-276.7%
A27_03	0.519	0.176	-66.2%	13.14	54.29	-313.1%
A28_01	0.500	0.204	-59.2%	14.40	22.01	-52.8%
A28_02	0.227	0.041	-82.0%	6.24	31.11	-398.5%
A58_01	0.837	0.258	-69.1%	9.69	44.65	-360.8%
A58_02	0.865	0.512	-40.8%	8.20	16.51	-101.4%
A58_03	0.892	0.528	-40.9%	9.18	14.64	-59.5%

8 Conclusions and discussion

This chapter contains the final conclusions and the discussion. In the conclusion the research question is answered. This is done per subquestion that was formulated, before answering the main question. In the discussion the limitations are discussed, together with recommendations for future work.

8.1 Conclusions

8.1.1 Best fitting machine learning technique

The first part of this research was focused on finding a suitable machine learning technique for traffic state estimation. In order to answer this, different models were tested on a dataset that was gathered on a motorway with much congestion, caused by a lane drop. Three kind of models were researched: regression models, decision tree learning models and neural networks. All those models were tested in a recurrent and a non-recurrent form.

The first important conclusion to draw here is that recurrent models are always preferred when compared to non-recurrent models. Recurrent models have the ability to make iterations, this makes is possible to take into account estimations of previous and future instances. Since traffic patterns are in fact sequences of the defining traffic variables, including estimations of those instances improve the estimation in the following iteration. In all tested models the recurrent version scored significantly higher than the non-recurrent version, both in congestion as in speed estimation.

Regression models gave satisfactory results in congestion estimation, but failed to capture the traffic patterns in speed estimation. In fact regression is not a machine learning technique, but a mathematical approach for an estimation model. Decision tree learning models performed better than regression models on speed estimation, while on congestion estimation the regression models scored a little better. Both these techniques can help in estimating traffic state, but is not the best option for doing this.

Recurrent neural networks provided the best results in the testing. In congestion estimation the RNN had an f-score of 0.90 on identifying congestion, which means that most congestion is detected and also there are not many false positives in congestion estimation. On speed estimation the RNN managed to estimate speed with a RMSE of 9.4 km/h. This means that the estimation is on average close to the real measured speed.

In the validation of these results on other road sections it was found that the RNN has a different performance on these road sections. It scored very well on road section with a clear identifiable bottleneck in which all traffic intensities were known. This occurred most often in the case of a lane drop. In the cases were a ramp is included in the road section, not all traffic flow was captured by the model. This is the most probable reason the model performed worse in these cases. In cases where there was no clearly identified bottleneck congestion estimation performed bad, while speed estimation scored good. The bad scoring on congestion estimation is caused by a lack of example data, since not much congestion was included in the data set. Speed estimation went better, because those road sections often have similar speeds, because of the lack of congestion.

8.1.2 Chosen input variables

In order to improve the results of traffic state estimation, three additional input attributes were tested on its influence on the model. Those three input variables that were tested are the influence of the weather (fog and rain), the influence of the share of long vehicles on the road, and the influence of adding the deviation in the intensity data of the input intensities. The results of this testing are compared to a base measurement. Testing was done by using the RNN that was tested best in the previous section.

The weather did not have a large influence on the estimation. Data about whether or not it rained and there were foggy conditions from a nearby weather station were added to the model. No improvement was found by adding these variables.

Adding the share of long vehicles as input variables was found to improve the model with 7.8% on the f-score in congestion estimation and 22.9% on the RMSE in speed estimation. The improvement on the speed estimation can be explained by the fact that long vehicles often have a lower speed than short vehicles. Adding information on the length of vehicles was found to improve the results of the model.

Also adding deviation in the intensity data was found to make improvements to the model. A higher value in the deviation can indicate a change in traffic patterns and thus give useful information to the model. The f-score improved by 10.1% by adding this information, and the RMSE improved with 19.1%. Because of these numbers adding deviation data was found to be useful to the model.

The deviation and the share of long vehicles were also added combined to the model. This resulted in a increase of the f-score of 10.5% and the RMSE improved with 28.6%. The combination was tested on the other road sections. It must be noted that for two road sections there was no vehicle length information available. On these two road sections adding the additional attributes resulted in worse scores. On the other roads there was found a small improvement on the performance measures on average. Although it must be said that the improvement was in general smaller than the improvement measured on the tested road section.

8.1.3 Approach for congestion and speed estimation

Two approaches for a merged dataset have been tested. In the first place all the train sets of all road sections were added and a model was trained on this and tested on the separate road sections. In the second place a merged train set was created in which the to be tested road was not included. In order to do so additional information on the characteristics of the road sections was added and all intensities were normalized to a percentage of the capacity of the specific road section, this was done for both tests.

The first test provided results that were comparable to the tests that were carried out on individually trained road sections. On average the scores were a few percent worse, but for others there was no difference or they even scored a little better. This shows that training on many different road sections still means that it is possible to make estimations for single road sections. The model does not get 'confused' to a great extent by adding varying data. The model is still able to use the instances that are needed for classification.

When data is tested on a model that is trained on data in which the tested road section is not included, the results drop enormously. Even though some comparable road sections score a little better than other road sections, there is no indication that the model can transfer characteristics of one road to another. In order to get good results on the test data, there must be data from the same road section in the training data.

From these results can be concluded that the RNN model is incapable of extrapolating. Interpolation shows slightly better results, but in general the model fails here too. With all attributes that were provided in this research extrapolation an interpolation are not

8.1.4 Use of machine learning in traffic state estimation

The main question for this research is 'How can machine learning techniques estimate traffic state based on intensity data?'. This question was split up in the three parts that are discussed above. These three parts together answer the main question. The results of the tested road section on the A58 near Tilburg are shown in table 8.1.

For estimation of traffic state (congestion state and speed) recurrent neural networks are useful. Those networks are to be fed with intensities of the road section, a location upstream and downstream and of time intervals just around the researched instance. Especially in case of a bottleneck and in cases where no intensities are lost or added by ramps, RNNS can estimate traffic state based on intensities. Adding

information on vehicle length and deviation of the intensity data help to improve those estimations.

In order to give a good estimation of traffic state on the road sections it is important that data from that road section is included in the training set of a model. The best option is to have a model for each road section separately, but a trained set on a combination of road section will also give satisfactory results. It is of no use to estimate traffic state by a model not trained on the specific road section, such models have no estimation power.

TABLE 8.1: Summary of all results on A58 near Tilburg.

Test	F-score	RMSE
RNN, tested on the same road section as it is trained on.	0.783	11.49
RNN, tested on the same road section as it is trained on, with additional attributes.	0.865	8.20
RNN, tested on a model trained on multiple road sections, including the tested road section, with additional attributes.	0.735	12.20
RNN, tested on a model trained on multiple road sections, not including the tested road section, with additional attributes.	0.512	16.51

8.2 Discussion

In the discussion the practical use of the research is discussed, but also its limitations are discussed. This discussion finishes with recommendations for future work on the topic.

8.2.1 Practical use of findings

The outcomes of this research can be useful in a number of situations. The context of this research was finding corresponding speeds for the modelled intensities of INWEVA data. This research did not use modelled data, but used measured data, but this does not mean it has no meaning for modelled data. RNNs can be useful in this context. A model trained for a specific road can be applied to modelled data. Of course, this model will have to be validated by a set of modelled or known speeds, connected to the modelled intensities.

The fact that it was found that training data from the same road section is needed that this model will be applied to, makes that these findings cannot be used for INWEVA right away. It was desired that a model that was trained on known intensity-speed combinations could give good estimations on other road sections where only modelled intensity data was known. Because of the fact that neural networks are known for their limited capability of extrapolating and interpolating, it is expected traffic state estimation using neural networks that are trained on other road sections than at which they are tested, will always remain a difficult task.

Possible directions of accomplishing estimation of road sections outside the training set can be sought in adding more input attributes or extending the train set. In this research all attributes that were expected to have an impact on the outcome – this are the attributes that define the characteristics of a road section – were added as input attributes. Therefore giving directions in which can be sought for other input attributes that may capture all characteristics of a road section is difficult. Possibly a research on the impact of the attributes that were already selected can give directions. Extending the training set to more road sections may result in grasping a more complete picture of the Dutch motorway network. By attempting to get a more extensive input of road section the results might improve. However, a drawback will be more calculation time for all new instances added to the model. Also, even though it may lead to an improvement of results, it is not expected capture all characteristics of road sections not in the training set.

Another use case for this research are motorways on which only intensities are measured. If only intensities are measured, but information on speed is desired, a RNN can be trained on such motorways. In order to do this for a certain period speeds will need to be registered. This can optionally be done by

using floating car data, which can give relatively precise data on speeds. Those speeds can be used to train a RNN that can be used for further estimations on the speed, without having access to information on speed. A drawback for this is that this cannot be applied real time, because the RNN needs to have the input of a time series, that also lies in the future.

8.2.2 Limitations

The research that has been done has limitations that are discussed here. Most of the limitations are due to the chosen methodology, while other thing are limited by the properties of the available data.

The first limitation of this research is that the followed approach does not make it possible to make real time estimations or predictions. Because future data is used as an input to the model, real time prediction is impossible. Also applying RNNs, in which future estimations are available in every iteration makes it impossible to make estimation for the future. These choices limit the applications of this research. If real time estimations or predictions were possible, the applications could be much broader.

A limitation because of a lack of data is that on most ramps there was no intensity data available. Because of the lack of data on ramps not all intensity data that plays a role on the specific instance was captured, the model cannot make a good estimation of difference between the measured location and the up– or downstream location, because a certain number of vehicles have left or entered the motorway. In stationary conditions this number can be obtained because of the law of conservation of flow. The number of vehicles that take the ramp is then equal to the difference of the location upstream and downstream of the ramp. But in conditions at which speeds are different, this number cannot be determined. It was hoped that this lack of information did not matter much, because the model might have been able to capture this by itself. This however did not work very well, because the performances near ramps was worse than at other locations.

By choosing only twelve road sections this research does not give an overview of the whole motorway network of the Netherlands. Also complex locations as motorway intersections or locations with a plus– or peak lane were not included in this research. The reason for this is that those locations would not fit in the format of data that is used in this research. For example at intersections there is not one upstream or downstream locations, but there are multiple. Or in the case of a peak lane the number of lanes would vary over the day and it is not exactly known when those lanes were open or closed. The choice not to include such locations limits the applications for this research.

A drawback of the outcome of this research is that it does not give new insights in the relation between intensity and speed. The neural network that was used is a black box and it remains mostly unclear why certain estimations are made. Only the quality of the result can be used as an output. In other words the neural network shows that a relation can be found between speed and a pattern of intensities, but how this relation works is unclear. The DTL and regression models give more insight in what happens in the model. This however also does not tell much about the speed intensity relation. The regression model only shows which inputs have a positive and negative correlation to the output and to what extent. The DTL gives a tree of rules, which in this case is so extensive, it does not really give much insight. In addition, the DTL and regression model performed worse than the RNN, meaning that the relation that eventually could be found is less reliable.

8.2.3 Future work recommendations

This research has some points on which further research can be of added value. A few possible research directions for future research are outlined.

An option for further research is applying neural networks on the relation between modelled data and speed. A data source for speeds can be floating car data. It would be interesting to see whether or not the same relations between those two data types can be found as were found in this research. If in that case the same results are found, models can be trained for estimating speed based on modelled intensity data.

This researched focused on estimating speeds when only intensities are known. Researching the other way round could be interesting as well: Estimating intensities based on speeds. On motorways where there is no observation of traffic this could give insight in the number of vehicles using that specific road. Speeds can be used from external parties, again floating car data is the most likely data source for this. Also, if it is possible to estimate intensities based on speed, with the same results as the RNN has in this research, the relation between speed and intensity can be explained better.

Finally, the last suggestion for further research is making speed estimation real time or even for the future. In order to do so intensity values in the future can no longer be used. For making prediction for the near future, even the intensity value of the present can no longer be used. Making such a model can be of great use, because on the short term congestions or other disruptions can be predicted. This could make improvement for example in software for navigation. Making predictions for the future will be hard however. In the models that are used in this research the present value for intensity is very important for the estimation. The future value for the instance is less important, but is also of importance in speed estimation.

References

Alpaydin, E. (2010). *Introduction to Machine Learning* (Second ed.). Cambridge, Massachusetts: The MIT Press.

Altintasi, O., Tuydes-Yaman, H., & Tuncay, K. (2017). Detection of urban traffic patterns from Floating Car Data (FCD). *Transportation Research Procedia*, *22*, 382–391. Retrieved from http://dx.doi.org/10.1016/j.trpro.2017.03.057

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.

Blandin, S., Salam, A., & Bayen, A. (2011). Individual speed variance in traffic flow : analyse of Bay Area radar measurements. *91th Annual Meeting of the Transportation Research Board*(12).

Bulteau, E., Leblanc, R., Blandin, S., & Bayen, A. (2012). Traffic flow estimation using higher-order speed statistics. , *7*.

Coifman, B. (2001). Improved velocity estimation using single loop detectors. *Transportation Research Part A: Policy and Practice*, *35*(10), 863–880.

Coifman, B., & Kim, S. B. (2009). Speed estimation and length based vehicle classification from freeway single-loop detectors. *Transportation Research Part C: Emerging Technologies*, *17*(4), 349–364.

DatMobility. (2017). *Rapportage verkenning ANN MISS Classificatie van stagnatie* (Tech. Rep.).

De Jong, A. J. (2012). *Quality of Real-Time Travel Time Information* (Doctoral dissertation, Universiteit Twente). Retrieved from https://www.utwente.nl/ctw/vvr/education/Master/finished_graduation_projects/afstudeerders_per_jaar_2/pdf/2012_09_26_Joost_de_Jong.pdf

Fusco, G., Colombaroni, C., & Isaenko, N. (2016). Short-term speed predictions exploiting big data on large urban road networks. *Transportation Research Part C: Emerging Technologies*, *73*, 183–201. Retrieved from https://www.sciencedirect.com/science/article/pii/S0968090X16302121

Goodwin, L. (2002). Weather impacts on arterial traffic flow. *Mitretek Systems Inc*, 4–8. Retrieved from http://ops.fhwa.dot.gov/Weather/best_practices/ArterialImpactPaper.pdf

Irawan, M. Z. (2010). Implementation of the 1997 Indonesian Highway Capacity Manual (MKJI) Volume Delay Function. *Journal of the Eastern Asia Society for Transportation Studies*, *8*, 350–360.

Jain, M., & Coifman, B. (2003). *IMPROVED SPEED ESTIMATES FROM FREEWAY TRAFFIC DETECTORS* (Doctoral dissertation, The Ohio State University). Retrieved from http://www2.ece.ohio-state.edu/$\sim$coifman/documents/ImprovedSpeed.pdf

Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, *54*, 187–197. Retrieved from https://www.sciencedirect.com/science/article/pii/S0968090X15000935

MathWorks. (2017). Deep Learning with Matlab. *Introducing Deep Learning with MATLAB*, 15. Retrieved from https://fr.mathworks.com/content/dam/mathworks/tag-team/Objects/d/80879v00_Deep_Learning_ebook.pdf

NDW. (2013). *Samenvatting rekenregels historische data.* Retrieved from https://www.ndw.nu/downloaddocument/447e721d82e44392f417d022416b4c68/RapportNDWRekenregels2013-samenvatting.pdf

NDW. (2018). *Historische open data aanvraag periode.* Retrieved from http://83.247.110.3/OpenDataHistorie

Ng, A. (2018). *Machine Learning.* Retrieved from https://www.coursera.org/learn/machine-learning/home/welcome

Nielsen, M. (2017). *Neural Networks and Deep Learning.* Retrieved 2018-03-14, from http://neuralnetworksanddeeplearning.com/index.html

Polson, N. G., & Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, *79*, 1–17. Retrieved from https://www.sciencedirect.com/science/article/pii/S0968090X17300633

Python-weka-wrapper. (2019). *Python wrapper for the weka machine learning workbench.* Retrieved from https://pypi.org/project/python-weka-wrapper/

Quiza, R., & Davim, J. P. (2009). *Computational Modelling of Machining Systems* (No. May 2015).

Rijkswaterstaat. (2018). *INWEVA verkeersintenstiteiten.* Retrieved from https://nis.rijkswaterstaat.nl/portalcontent/logon/p2_33.html

Seo, T., Bayen, A. M., Kusakabe, T., & Asakura, Y. (2017, jan). Traffic state estimation on highway: A comprehensive survey. *Annual Reviews in Control*, *43*, 128–151. Retrieved from https://www.sciencedirect.com/science/article/pii/S1367578817300226

Soriguera, F., & Robusté, F. (2011). Estimation of traffic stream space mean speed from time aggregations of double loop detector data. *Transportation Research Part C: Emerging Technologies*, *19*(1), 115–129. Retrieved from http://dx.doi.org/10.1016/j.trc.2010.04.004

Stewart, M. (2019). *The actual difference between statistics and machine learning.* Retrieved from https://towardsdatascience.com/the-actual-difference-between-statistics-and-machine-learning-64b49f07ea3

Treiber, M., & Kesting, A. (2013). *Traffic Flow Dynamics* (Vol. 2013). Berlin, Heidelberg: Springer-Verlag. Retrieved from http://link.springer.com/10.1007/978-3-642-32460-4

University of Waikato. (2019). *Weka 3: Machine learning software in java.* Retrieved from https://www.cs.waikato.ac.nz/ml/weka/

Wang, Y., & Papageorgiou, M. (2005). Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. *Transportation Research Part B: Methodological*, *39*(2), 141–167. Retrieved from https://www.sciencedirect.com/science/article/pii/S0191261504000438#BIB19

Zhu, J. Z., Cao, J. X., & Zhu, Y. (2014). Traffic volume forecasting based on radial basis function neural network with the consideration of traffic flows at the adjacent intersections. *Transportation Research Part C: Emerging Technologies*, *47*, 139–154. Retrieved from https://www.sciencedirect.com/science/article/pii/S0968090X14002010