Master thesis

Methods for Short Term Prediction of Train Delays for Different Operating Schedules

Author:
M.A. Albers
s4455843

Supervisor Radboud University:
Dr W. Bosma

Supervisor Nederlandse Spoorwegen:
P.J. Fioole

March 2021

Preface

Before you lies the master thesis which is written to complete the master Mathematical Foundations of Computing Science at Radboud University in Nijmegen. This thesis is the final result of the time I spent researching at the innovation department within the Nederlandse Spoorwegen. During this time the Covid-19 pandemic started, which resulted in me working at home instead of at the office. This proved to be an environment in which I do not flourish, but it made me learn a lot about myself and what kind of job I want. Even though this internship did not go as expected, I look back on it as a great learning experience and I had a great time with my colleagues at the Nederlandse Spoorwegen.

First I want to thank my supervisor at the Nederlandse Spoorwegen, Pieter-Jan Fioole. You were very patient with me and helped me out whenever I got stuck. I enjoyed our discussions when the results got strange, trying to explain why they would be logical. Next, I want to thank my university supervisor Wieb Bosma. Even though you are not an expert on the subject of this thesis, you were always willing to give advice and ask the right questions to help me get back on track. Both my supervisors were always prepared to make time for me, whenever I had a question.

I also want to thank my family. My parents for providing me with material, creating a great work environment when needed and listening to my problems even though you did not understand the subject. My brother Stan Albers and my sister Renée Albers I want to thank for sharing their experience in writing a thesis and giving a critical look on mine. Without my boyfriend Paul Hunink, this thesis would not have been finished. Thank you for being there during all the ups and downs, never doubting me and encouraging me whenever I got stuck.

I want to thank everyone who took their time to listen to my problems and who gave me advice. Especially Henk Don for confirming I was in the right direction and showing interest in my project. Also Abel Vleeshouwers deserves a thanks on this page, for reviewing my thesis and giving notes on spelling and grammar mistakes.

Margot Albers, March 2021

Abstract

Providing accurate information regarding the arrival times of trains is valuable for costumer satisfaction. Passengers want to be kept up to date on their travels, and eventual changes in arrival time are important. Therefore it is important to have a good model that predicts the delay of trains. There have been previous projects at the Nederlandse Spoorwegen regarding this problem, where one intern, Leonieke van den Bulk, created a model using machine learning techniques.

During the internship a global pandemic occurred. As only essential workers were allowed to travel by trains, the Nederlandse Spoorwegen drastically changed their operating schedule. This new schedule is called the basis schedule and it is a reduced version of the regular schedule. Because the operating schedules had never been changed this much before, this led to a unique experiment. Two methods for creating a model predicting the delay of trains were used. The first method used Neural Networks to train the data, which was copied from Leonieke van den Bulk's thesis. The second method used the frequencies of situations to calculate conditional probabilities. For every combination of features the best performing model was chosen. For both schedules the best combination was to only look at the delay 20 minutes ago.

For every method and schedule a model was created. So in total there were four different models. The models were tested on both the test set from the basis schedule and the test set from the regular schedule. This to answer the question whether there is a difference in predicting the delay of trains for the regular schedule and the basis schedule. The models trained on the regular schedule were better at predicting the test set from the regular schedule than the models trained on the basis schedule and vice versa. This is logical as the models were trained on the respective schedule, but the difference between the performances was smaller than expected. This showed that for a small difference in schedule it is not needed to train the models all over again, because for a large difference in schedule the models did not significantly outperform the other.

Also the performance of each of the two methods was compared against the other to see which method outperforms the other. The performances were very similar, but the conditional probabilities method had the best performance score overall. Furthermore, this method is intuitive and satisfies the wishes of the Nederlandse Spoorwegen more. Therefore the advice of this thesis is to stop researching Neural Networks for the prediction of delay.

Contents

1 Introduction

1.1 Problem

The Nederlandse Spoorwegen (called NS from here) is the main operator of passenger trains in the Netherlands. In 2019 it coordinated around 6000 trains every day, making sure more than 1.3 million passengers arrived at their desired location. In that year 62% of the population of the Netherlands made use of the services provided by the NS. The Dutch rail network is the busiest and densest in Europe and most of the trains on this network are passenger trains. Because of the tightness of the operating schedule, a small delay of a single train can cause delay to a large group of other trains.

For the NS it is important to provide accurate information on the arrival time of trains. Traffic control uses this information so that future stations can take precautions when there is a delay. Moreover, the expected arrival times are displayed on the information boards on stations, the website and the NS mobile application, so passengers know what to expect regarding their travels. This is why real-time data on the performance of trains regarding the schedule is collected. This data can be used to create models for predicting delay.

To predict delay, you need to think about different features and methods. Examples of features are the hour the train is moving or the delay of the train ahead. The NS has tried several methods for predicting delay, but the one currently in operation is the model that predicts that the future delay will be the same as the current delay. It is a simple but fair method since more than 90% of trains arrive on time. Previous theses by Leonieke van den Bulk and Eva Lehkà have aimed to improve on this method by creating a prediction model using machine learning techniques. The results seemed promising, as they slightly outperformed the current method.

However, there were some setbacks with the new models. When the results were shown to other departments within NS, the reactions were less enthusiastic. The new model has fewer wrong predictions in general, but false positive results (falsely predicting an increase in delay) occur. By construction, the current method has no false positive results. Customer satisfaction will decrease when these situations arise. For example, passengers could see the predicted delay, go to the store at the station to get a cup of coffee and then miss their train. They would have been on time if there was no predicted delay. Therefore, the departments preferred the current method. Furthermore, the new models have been tested on live data to see how they perform. The results were not as promising as the results in the theses.

All these setbacks cast doubt on the believe that when predicting the delay of trains a model created using machine learning techniques is the best method. One of the goals of this thesis is to find out whether these models are good methods for predicting delay and whether there is a more intuitive method that performs better.

1.2 Situation

This research took place during an abnormal situation: the COVID-19 pandemic. The government of the Netherlands issued a lockdown, which meant everyone had to stay inside and only essential workers were allowed to use public transport. Passengers were obliged to wear facial masks in trains and because everyone had to keep distance, less than half of the seats were available. This resulted in a large decrease of passengers for the NS, so a different schedule was created, called the basis schedule. This schedule is a reduced version of the regular schedule. For example, in the basis schedule all the intercity's, trains that only stop at important stations, were removed, shortened or scheduled to stop at every station. As there are a lot fewer trains riding in the basis schedule, one would expect the impact of a delay of a train on other trains to be less.

These two different schedules provided for a unique experiment regarding predicting the delays of trains. For each type of method two different models will be created, one trained by the data set containing the basis schedule and the other the regular schedule. Then these models will be tested on how they perform on trains from their own schedule, but also on trains from the other schedule. One of the goals of this thesis is to test if a model based on the basis schedule is significantly different from the regular schedule.

1.3 Research Questions

Combining the two goals described above, the main research question this thesis aims to answer becomes

What is the best method for predicting the delay of trains for different operating schedules?

Other research questions have been formulated:

- Is there a significant difference between creating a model for predicting delay for the basis schedule and for the regular schedule?
- Do Neural Networks outperform a model based on classic statistical methods?

To answer these questions, this thesis will look at two types of models that predict 20 minutes into the future. One model uses Neural Networks and is based on a model described in the thesis of Leonieke van den Bulk. The other model will be created using conditional probabilities. The two different schedules and the two types of models will be looked at and compared. The thesis will follow Figure 1.

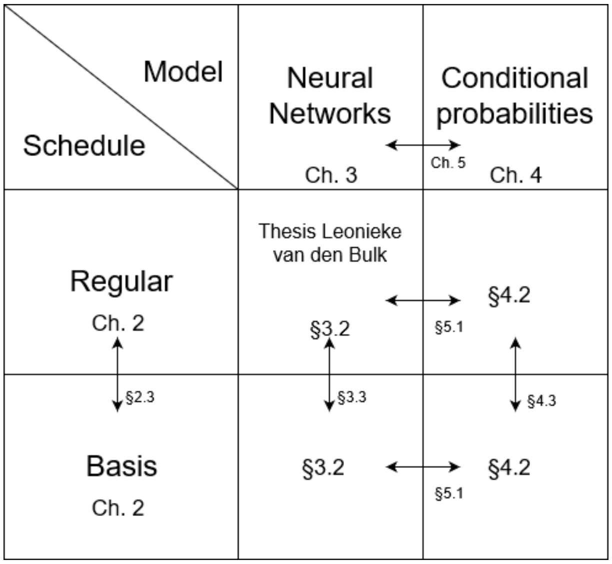

Figure 1: Overview of thesis

1.4 Scope

This research focuses on short term train delay prediction, because of the use of the model described in Leonieke van den Bulk's thesis. This model predicts the delay of trains twenty minutes into the future. To better compare the methods of creating models, all the models that will be researched in this project predict the delay twenty minutes into the future.

2 Methods and data

In this section, background information is given on predicting the delay of trains at the NS. First, we are going to look at how the NS currently predicts the delay of trains. Then we take a closer look at the data that is used to predict the delay currently, and is used in the models created in Sections 3 and 4. Because it is important to see that the basis schedule and the regular schedule are significantly different for the experiment, the content of the data sets of the two schedules are compared to each other. At the end of this section, the method for calculating the performance of the predictors is presented.

2.1 Current method for predicting delay

The NS has the system *InfoPlus* that processes logistics information and provides travel information for the passengers. *InfoPlus* consists of three subsystems HARM, CRIS and PUB. In HARM different information is harmonized such that it is usable input for CRIS. This information consists of, for example, the annual planning, the day planning and the data from the train tracking system at ProRail. The latter is the data that is used to create the models in Sections 3 and 4, and is explained in Section 2.2. CRIS then transforms this harmonized information into current travel information, including the predictions of the arrival times of trains. So in subsystem CRIS the method for predicting delay is built. Next, PUB publishes this travel information to the information boards on the stations and other media.

The method that currently predicts the delay of trains at the NS is seen in Figure 2. When there is no delay the prediction will be that the same holds for the future stations. When there is a delay, the method predicts that the train can catch up a fixed percentage of the travel time between stations. Also, when a train has scheduled for a stop of longer than two minutes at a station, the stop is reduced to two minutes. The delay is then reduced by the extra time.

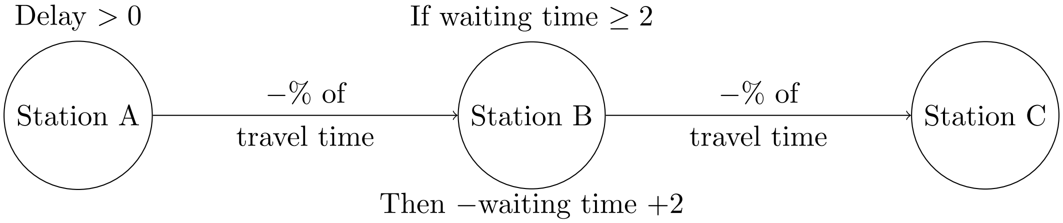

Figure 2: Current method for predicting delay.

This method works fairly well, as the travel schedule is created in a similar way. Between every station, they add the same fixed percentage to the actual travel time. So when there is a delay the train can make up for lost time. But on the other hand, at some stations there is necessary extra time planned. For example, when the train changes direction and the train driver has to move to the other side of the train or when a trainset is added to the train. Then the reduction of the extra time from the delay is inaccurate.

In practice, this method predicts that when there is a delay the train will catch up one minute of delay in twenty minutes. The current method will be referred to as this, because the models created in this thesis will predict the delay twenty minutes into the future.

2.2 Content of the data sets

As seen in the section above, ProRail provides data for *InfoPlus* of the current situation on the Dutch rail network. For this research we are taking a look at two data sets resembling the two schedules. The first data set contains all the trains that travelled between 16-12-2019 and 23-02-2020, when the NS ran the regular operating schedule. The second data set concerns the ones between 23-03-2020 and 28-04-2020, when there was the basis operating schedule. The data is created by measuring points on the railway. When a train passes such a point, it will create a new data point. All of the stations are measuring points, but there are also a few in between. Two examples of data points twenty minutes apart are given below.

Travel Date	**Series**	**Characteristic**	**Number**	**Location**	**Activity**
2019-12-16	5400E	SPR	5462	Zvt	V
2019-12-16	5400E	SPR	5462	Hwzb	K_V

Planned time	**Realisation**	**Delay**
2019-12-16 19:34:00	2019-12-16 19:34:49	0
2019-12-16 19:54:36	2019-12-16 19:57:41	3

Original plan	**Delay Jump**	**Cause**
2019-12-16 19:34:00	0	-
2019-12-16 19:54:00	2	Other train

Features

All the data points give information on the different features. 'Travel Date' is different from a normal date, as it doesn't start at 00:00 but at around 5:00 in the morning and ends at around 03:00 the next day. This is because a train can start their route before midnight and end it after midnight. A 'Series' describes all the trains riding the same route, with the letter *E* or *O* at the end to indicate the direction. The 'Characteristic' describes what type of train it is, as for example an Intercity (IC) only stops at the important stations it passes and a Sprinter (SPR) at every station. The 'Number' of a train indicates a specific train ride of that day. It indicates the series, direction and order in departing time of trains travelling the same route. The 'Location' indicates the place the train has passed a measuring point. 'Activity' describes what the train did at the measuring point. For example, *V* stands for leaving the station after a stop. 'Planned time' and 'Original plan' are often the same, as they both stand for the time the train is planned to arrive at the measuring point. But sometimes a short-term alteration of the plan can be made. 'Realisation' indicates the time the train passes the measuring point. 'Delay' is the difference between 'Realisation' and 'Planned time' in minutes. 'Delay Jump' is the difference in delay in comparison to the previous measuring point. When there is a jump in delay, most of the time a cause of this delay has been given which is indicated by 'Cause'.

Change of plan

So sometimes the plan is changed by traffic control. For instance, a train has a delay and when the planned arrival time is altered to be later, it causes the barrier at a railroad crossing to drop later. Ohterwise, the cars should wait until the train has arrived. As the delay of the train is based on the difference between realisation and planned time, this could cause weird effects in the data. For example, a train has thirteen minutes delay at a data point and it takes six minutes to arrive at the next data point. Then when traffic control changes the schedule, the delay at the next data point could for instance now be one minute based on the new plan. But the train would never be able to catch up twelve minutes of delay in a time span of six minutes. From a passenger point of view, the delay has not been changed. But from a plan point of view, the delay has. Basing the delay on the difference between the original plan and realisation also causes weird effects in the data, because the train driver will now drive according to the new schedule. He or she will not try to catch up with the original plan. So there is not one good method to base this delay on, but it is important to know the faults of both.

Train series

As explained before, a train series describes all the trains travelling the same route. There are more than 90 train series currently active on the tracks. As the progression of delay is different for each train series, the models are trained for each train series separately. Training 90 models is a lot of work, therefore the choice has been made to look at four train series: 3000, 4000, 4400 and 4900. The 3000 series is the same series as tested in the thesis from Leonieke van den Bulk. It is the intercity from Nijmegen to Den Helder, it stops at the main stations. In the basis schedule, it stops at every passing station. The rest of the series are all Sprinters, which means they always stop at every passing station. The 4000 series starts in Uitgeest and ends in Rotterdam Centraal, the 4400 starts in 's Hertogenbosch and ends in Deurne and the 4900 starts in Almere Centrum and ends in Utrecht Centraal.

Splitting the data sets

Both the data sets are divided into three separate sets: a training set, a validation set and a test set. The training set is used to set the right parameters of the models so they fit the training set. The validation set makes sure the models are not overfitted on the training set. Overfitting means that a model is too good at predicting a certain data set, but does not perform well on future data points. The performance of the models is checked on the validation set and then the parameters or features that best fit the validation set are chosen. In the end, the performance of the models is tested on the test set. For the basis schedule, the training set contains the train rides that occurred between but not including 22-03-2020 and 23-04-2020, the test set on 23-04-2020 and the validation set between but not including 23-04-2020 and 29-04-2020. For the regular schedule, the training set consists of the train rides between but not including 15-12-2019 and 05-03-2020, as the test set the train rides on 05-03-2020 and as the validation set the train rides between but not including 05-03-2020 and 23-03-2020. The day that is chosen to be the test day is Thursday for both schedules. In Table 1 the number of data points per data set can be found.

Schedule	Series	Training set	Validation set	Test set
Regular	3000	404606	91237	5324
Basis	3000	144175	23954	4512
Regular	4000	383692	84127	5027
Basis	4000	151723	24806	5106
Regular	4400	139605	30387	1910
Basis	4400	50824	8674	1921
Regular	4900	88245	20258	1342
Basis	4900	38773	5395	1400

Table 1: Number of data points per data set.

2.3 Difference between the schedules

In this subsection we explain the use of the two distinct data sets, as we have one containing data points during the basis schedule and on during the regular schedule. It hardly ever occurs that such a drastic new schedule is created to operate the trains for more than a month's time. This means that in this thesis there is going to be a unique experiment on the two different schedules. For each method of creating a model, two models are created. One is trained on training data from the basis schedule and the other on the regular schedule. So we have a basis model and a regular model per method. For each of the schedules a portion of their data set is taken to create a test set, so a basis test set and a regular test set. Then for each model the performance on both test sets is calculated. This will answer the first research question: "Is there a significant difference in creating a model for predicting delay for the basis schedule and the regular schedule"? One would expect that a model trained on the basis schedule is better at predicting the basis test set than a model trained on the regular schedule. For this experiment to go well, the schedules have to be significantly different.

In Figure 3 you can see the distributions of the number of minutes delay per schedule. For both schedules between 75% and 90% of the data points have zero minutes delay, so zero minutes delay occurs most often. This means that the largest part of the data sets is one value, which makes it hard to predict the other values. Predicting that a train never has any delay therefore is already a good predictor.

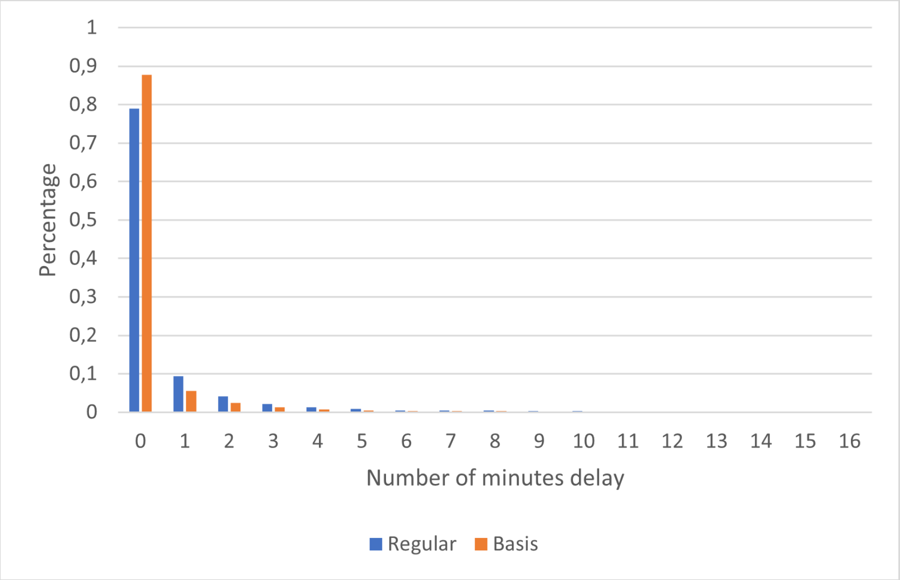

Figure 3: Distribution of basis schedule and regular schedule.

In Figure 4 you can see the statistics of the two different schedules compared to each other. The statistics include the average μ, the standard deviation, the variance σ^2 and the number of large increases per train ride. The average delays both are between zero and one, but the average delay for the regular schedule is twice as big as the average delay for the basis schedule. The variance is the measurement for the spread of the numbers in the data set. The formula for calculating the variance is given in Equation 1, where n is the total number of data points and x_i the delay of a data point.

$$\sigma^2 = \frac{1}{n}\sum_{i=0}^{n-1}(x_i - \mu)^2 \tag{1}$$

The variance looks at how far each data point is from the average μ, so when the variance is large it means that the values are more spread out. Looking at Figure 4, this means that the data points in the regular schedule are more often further away from it's average than the data points in the basis schedule. The standard deviation is computed by taking the square root of the variance, so it also is a measurement for the spread of the data. Large differences between data points and the average result in a higher standard deviation. In this case, it is larger for the regular schedule than for the basis schedule. The number of large increases per train ride calculates the times a train made a jump of three minutes in delay in comparison to the previous data point and divides it by the number of train rides. This is also twice as large for the regular schedule as for the basis schedule. This could be caused by the higher number of trains on the tracks in the regular schedule. When it is busier and a train has a delay, the probability of causing delay to another train is higher.

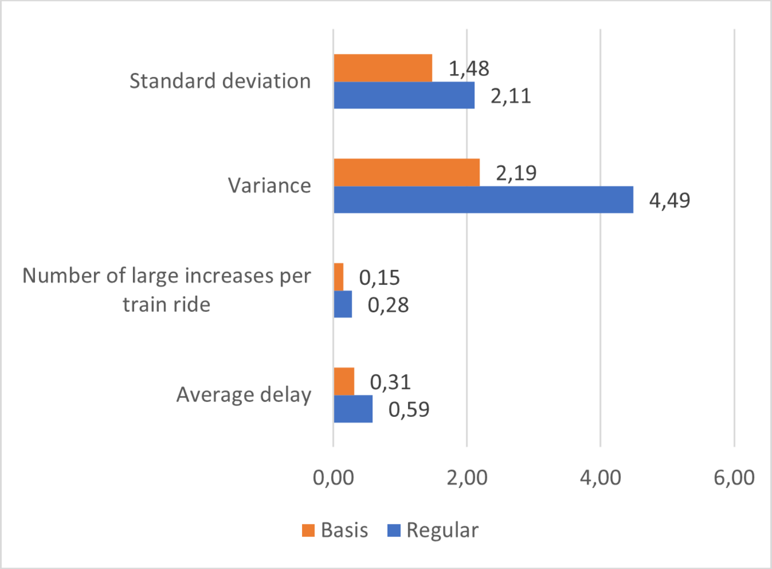

Figure 4: Statistics of the basis and regular schedule.

These statistics show that the schedules have significantly different distributions.

2.4 Performance score

In this thesis four models are created by using two different methods and two different data sets. All these models predict a numerical value, so they have a 'Regression' label. The performance is calculated in the same way as in the thesis of Leonieke van den Bulk [1], using the root mean squared error (RMSE). Van den Bulk also created models using other labels, but those will not be considered in this thesis. In Chapter 3.2 it is explained why. The root mean squared error is given in Equation 2 where D is the number of data points, $\hat{x}_d$ is the prediction of data point d and x_d is the true value.

$$\text{RMSE} = \sqrt{\frac{\sum_{d=1}^{D}(\hat{x}_d - x_d)^2}{D}} \tag{2}$$

This scoring function implies that the lower the score, the better the performance. It punishes large differences between the prediction and the true label harder than small differences. For example, when a model predicts perfectly 99 times but one time it predicts with a difference of twelve minutes from the true label, it results in a higher score than predicting hundred times with a difference of one minute. The root mean squared error meets the wishes of NS, as large differences in delay are considered worse than small differences.

3 Neural Networks

In this section, the creation of the Neural Network model will be discussed and its results on the different test sets.

3.1 Introduction

First let us look at the idea of a Neural Network. In the theses by Eva Lehkà [2] and Leonieke van den Bulk [1] Neural Networks were used to create a model that predicts the delay of trains. A Neural Network is inspired by the human central nervous system. It takes a large set of data as training samples and develops a system that learns from these training samples. A Neural Network consists of layers: an input layer, hidden layers and an output layer. This is seen in Figure 5. The input layer x consists of input nodes that correspond to the features of the data. Examples of those features for our model are the day the train was travelling on or the number of minutes delay 20 minutes ago. The output layer y consists of output nodes, which correspond to the labels the data will be predicted to. The labels for our model are the numbers of minutes delay. The hidden layers $h_1, \ldots, h_k$ consist of neurons. The more hidden layers, the more complicated the Neural Network. Each input node is connected to each neuron in the first hidden layer by an arrow, each neuron in a hidden layer to each neuron in the next hidden layer, etc. Finally, each neuron in the last hidden layer to each output node.

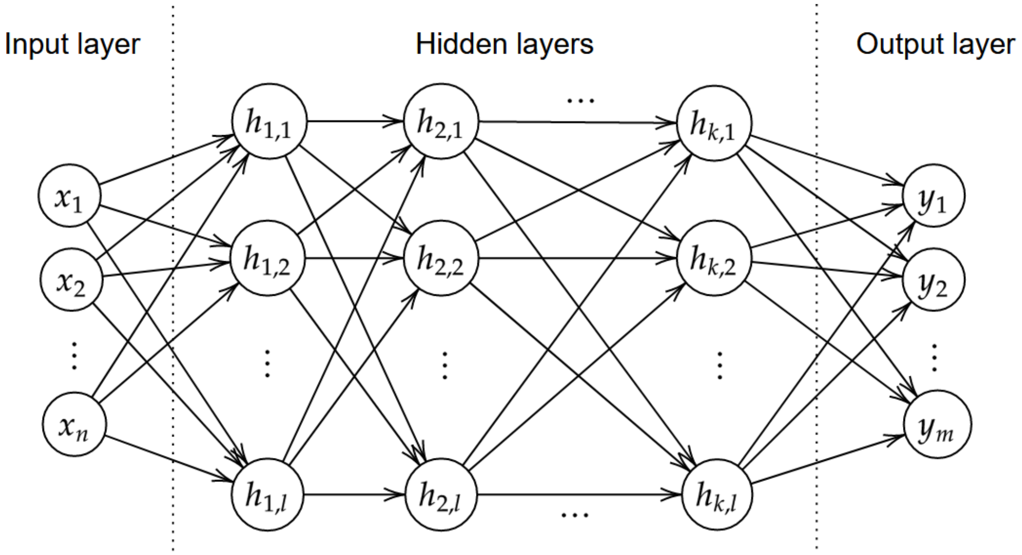

Figure 5: Overview of a Neural Network.

In Figure 6 the architecture of a single node can be found. The outputs of all the nodes in the previous layer become the inputs $i_1, \ldots, i_j$ of the node. Each arrow has a weight $w_1, \ldots, w_j$, real numbers expressing the importance of the inputs to the output of the node. The output o is computed using a series of actions. First, the weighted sum $\sum_d w_d x_d$ of the inputs is calculated. Then the bias b is added, which is a measure of how easy it is to activate a neuron. Finally, an activation function φ is used, which differs depending on what types of labels you want to predict. For example, for binary classification the sigmoid function $\varphi(z) = \frac{1}{1+e^{-z}}$ is used, which has as input a real number and as output a number between 0 and 1. The bigger the z, the closer the output is to 1 and the smaller z, the closer the output to 0. So the output of a node becomes the following function.

$$o = \varphi(b + \sum_{d=1}^{j} i_d w_d) \tag{3}$$

The output nodes compute in the same way their output, which represents the probability of that label being the true label. The computation of the output of the neurons and the output nodes is called *forward propagation*. The activation functions all have in common that a small change in weights or bias results in a small change in output. So a Neural Network can change these weights w and biases b such that the probabilities of each data point receiving a certain label is highest at the true label. This is done by minimizing the cost function C, which is defined by:

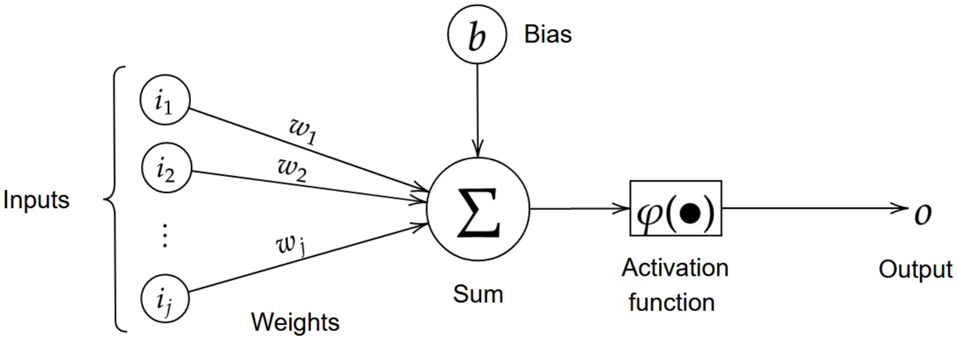

Figure 6: Computation of output of node.

$$C(w, b) := \frac{1}{D} \sum_x ||y(x) - a(x, w, b)||^2$$

Here w is the vector of all weights, b the vector of all the biases, D the number of training samples, x an input vector, $y(x)$ the vector representing the true label of x and $a(x, w, b)$ the vector representing the output of the network with input x. The value $C(w, b)$ is obviously nonnegative and the smaller $C(w, b)$, the closer the outputs of the Neural Network are to the true labels. The goal of training a Neural Network is to find w and b that minimize the cost function, given training samples. The weights are changed using gradient descent, where each weight's gradient is subtracted from the weight. The process of changing the weights and biases is called *back propagation*. A more detailed explanation of the algorithm is found in Rumelhart et al [6].

3.2 Model of Leonieke van den Bulk

The Neural Network used in this thesis is based on the code and thesis by Leonieke van den Bulk. In this paragraph, the model and what parts of it are being used in this thesis are explained. For a detailed explanation of the model, consider the thesis by Leonieke van den Bulk [1]. In Van den Bulk's thesis there were multiple models with different features which all performed roughly similar. As the aim of this thesis is to compare the model of Leonieke with statistical methods, the basic feature set that the thesis provided is used and no effort was put into optimizing the Neural Network model. The Neural Network model is recreated in the programming language Python.

The model predicts the delay of a train 20 minutes into the future, given the current delay. It has three different types of labels. The first is a binary label which represents a jump of five minutes or more (Yes/No), the second is a ternary label which represents a change of delay of at least two minutes (Decrease/Equal/Increase) and the last is a numeric label which represents the actual delay in minutes. The basic feature set consists of different types of features. The features ‘Current day of the week’ and ‘Current location’ are categorical variables, ‘Current hour’, ‘Current minutes’ and ‘Previous delay’ are numerical variables and ‘Direction’ and ‘Same train’ are booleans. Because of the features ‘Previous delay’ and ‘Same train’, two data points are needed: the current data point and the data point concerning the train twenty minutes before. The two examples from Section 2.2 become the training sample below.

Day	**Hour**	**Minutes**	**Direction**	**Location**	**Same train**	**Previous Delay**	**Labels**
1	19	54	1	16	1	0	0/2/3

The change in delay between the two data points is 3 minutes. The ‘Jump’ label becomes 0, as the change is less than 5 minutes. The ‘Change’ label will become 2 indicating ‘Increase’, as the change is bigger than 1 minute. The ‘Regression’ label is 3 as this is the delay.

Leonieke van den Bulk used rolling stock schedules from the NS in her code to create the model. It shows at which locations trains had to change their composition. Unfortunately, this is not available for the basis schedule, as the basis schedule was made in one week in contrast to a year for the regular schedule. In this thesis all the models will not make use of the rolling stock schedules, because this will result in a better comparison. Unfortunately, this has an influence on the prediction of delay. Van den Bulk’s code and data have been used to create a model for predicting delay to calibrate it to the results of Van den Bulk’s thesis. Also, a model is created without the use of rolling stock schedules, by tweaking the code a little bit. The results are seen in Table 2. As you can see the model created without rolling stock for the ’Jump’ label fails to accurately predict a jump in delay. The models using the ‘Regression’ label and the ‘Change’ label perform similarly to the results of Van den Bulk’s thesis.

	Jump			Change			Regression
	Precision	Recall	F1	Precision	Recall	F1	RMSE
Leonieke’s thesis	0,077	0,25	0,12	0,30	0,21	0,25	1,38
Code with rolling stock	0,05	0,25	0,083	0,29	0,24	0,26	1,39
Code without rolling stock	0	0	0	0,30	0,21	0,25	1,37

Table 2: Calibration of the code to Leonieke’s thesis.

In this thesis only the ‘Regression’ label is used, because this is the most practical label as it gives an actual number. Furthermore, the conditional probabilities model this Neural Network model is compared to only creates a numerical label. So for the other labels there is no other model to compare it to. Besides, the model with the ‘Jump’ label without rolling stock does not have similar results as Van den Bulk’s thesis.

3.3 Results

In this paragraph, the Neural Network model trained on the data from the regular schedule (regular neural model) and the Neural Network model trained on the data from the basis schedule (basis neural model) are tested on the two test sets. The performance of the models on the different train series and schedules are given in Table 3. The RMSE (see Equation 2) is used as the scoring function. As seen in Section 2.4: the lower the score, the better the performance.

The basis test set is easier to predict than the regular test set, as both models have a better performance score on the basis test set. However, the models perform significantly worse on the basis

Test set	Training set	4900	4400	4000	3000	all series
Basis	Basis	1,9741	0,9406	0,6280	0,6796	0,9326
Basis	Regular	1,9908	0,8802	0,7189	0,7128	0,9617
Regular	Basis	1,1299	1,3973	1,3622	1,3213	1,3301
Regular	Regular	1,0347	1,2959	1,2884	1,2697	1,2593

Table 3: Performance of Neural Networks models on different test sets.

test set containing the 4900 series than the regular test set containing the 4900 series even though both test sets have an average of around 0.52 minutes delay. The first test set has fewer jumps of more than three minutes delay than the second test set, but the largest delay that occurs is 22 minutes and 6 minutes respectively. Because the RMSE punishes large delays, the performance score is worse on the basis test set containing the 4900 series.

In almost all cases the models are better at predicting the test set from the schedule they are trained on, than the other model. Only at the basis test set containing just the 4400 series has the regular neural model a better score than the basis neural model. This is caused by there being slightly more jumps in delay on this day than normal. This fits the regular model better and therefore it has a better performance score on this set.

The difference between the overall performance scores of the models on the basis test set is 0.0291. The difference in the regular test set is a bit larger, namely 0.0708. These are small differences and so it is hard to say whether a model trained on a schedule performs better on the test set of that schedule than the model trained on the other schedule. Therefore there is not a large difference in creating a model using Neural Networks for the regular schedule and the basis schedule.

4 Conditional probabilities model

In this section we create two models based on conditional probabilities, one for the basis schedule and one for the regular schedule. Conditional probabilities are chosen to be the classic statistical method for creating a model predicting the delay of trains, because it looks at the frequencies of specific situations occurring. This makes the model created using this method an intuitive model.

4.1 Conditional probabilities

The data sets can be used to approach the probabilities of given events. In this thesis we desire calculating the probability of a train having a certain number of minutes delay. Let's say that feature A represents the number of minutes delay a train has, then $A = i$ is the event that a train has i minutes delay. This number i is rounded to a whole number. The probability of $A = i$ occurring can be approached by dividing the number of times a train has i minutes delay by the total number of data points. The more data points in the data set, the more accurate this probability can be estimated. The calculation is formulated as the following equation, where Ω stands for the set of all data points and $A(x)$ the number of minutes delay the train has in data point x.

$$P(A = i) = \frac{\#\{x \in \Omega \mid A(x) = i\}}{\#\Omega} \tag{4}$$

Now let's say the binary feature B_0 represents whether the train ride was on a weekday or in the weekend. Then $B_0 = 0$ and $B_0 = 1$ are the events that the train ride did not and did happen on the weekend, respectively. The conditional probability $P(A = i|B_0 = 1)$ represents the probability of a train having i minutes delay given the information that it's weekend. This conditional probability is calculated by dividing the probability of the events $A = i$ and $B_0 = 1$ both occurring divided by the probability of $B_0 = 1$. This can be simplified as follows.

$$\begin{aligned} P(A = i \mid B_0 = 1) &= \frac{P((A = i) \wedge (B_0 = 1))}{P(B_0 = 1)} \\ &= \frac{\frac{\#\{x\in\Omega|(A(x)=i)\wedge(B_0(x)=1)\}}{\#\Omega}}{\frac{\#\{y\in\Omega|(B_0(y)=1)\}}{\#\Omega}} \\ &= \frac{\#\{x \in \Omega \mid (A(x) = i) \wedge (B_0(x) = 1)\}}{\#\{y \in \Omega \mid B_0(y) = 1\}} \end{aligned} \tag{5}$$

Calculating the conditional probability now becomes quite easy. You divide the number of data points that both have i minutes delay and are in the weekend by the number of data points in the weekend. Adding k more features B_j and their corresponding events $B_j = m_j$ just results in the very specific event $(B_0 = m_0) \wedge \cdots \wedge (B_k = m_k)$. The conditional probability then becomes the following.

$$P(A = i \mid \bigwedge_{j=0}^{k}(B_j = m_j)) = \frac{\#\{x \in \Omega \mid (A(x) = i) \wedge (\bigwedge_{j=0}^{k}(B_j(x) = m_j))\}}{\#\{y \in \Omega \mid \bigwedge_{j=0}^{k}(B_j(y) = m_j)\}} \tag{6}$$

Given this knowledge of calculating conditional probabilities, conditional probabilities matrices can be made using frequency matrices. The frequency matrices are created by first taking a zero-matrix with size depending on which features are included and for A a numerical feature depending on how many minutes delay is included. For example, we take feature A with a minimum of 0 and a maximum of nine minutes delay, feature B_0 as defined before and feature B_1, which we will define as the location of the train, expressed in Utrecht and ¬Utrecht. So feature A is divided into 10 events, B_0 2 and B_1 2. Because A is the feature we want to predict, the columns in the matrix will correspond to the events of feature A. The rows will then represent all the possible combinations of events of B_0 and B_1. So the matrix will have size 4×10. Then for every data point, the values for each feature are checked. So for example, we have a data point that has two minutes delay, is located in Utrecht and is on a weekday. Then we add a one to the entry in the matrix representing these values. The creation of the frequency matrix is done when every data point is checked. The conditional probability matrix is then created by using Equation 6, which means for every entry in the frequency matrix the entry is divided by the sum of its row. The conditional probability matrix of the basis schedule of our example is shown in Figure 7.

	0	1	2	3	4	5	6	7	8	9
Weekday $\wedge$ Utr	0,835	0,074	0,034	0,013	0,009	0,012	0,005	0,003	0,003	0,011
Weekday $\wedge$ $\neg$Utr	0,791	0,098	0,044	0,022	0,013	0,007	0,003	0,004	0,004	0,015
Weekend $\wedge$ Utr	0,942	0,025	0,010	0,004	0,000	0,009	0,004	0,003	0,003	0,003
Weekend $\wedge$ $\neg$Utr	0,889	0,061	0,020	0,009	0,005	0,004	0,003	0,002	0,002	0,005

Figure 7: Example of a conditional probability matrix.

This matrix can be used to predict the delay of a train, by calculating the expected value of the rows. The expected value is the weighted average over the values, in this thesis's case the number of minutes delay. The weights are the probabilities of each value, as they all add up to one. The expected value of the feature A is defined in the equation below.

$$E(A) = \sum_i i \cdot P(A = i) \tag{7}$$

For conditional expectations only the conditional probabilities $P(A = i | \bigwedge_{j=0}^{k}(B_j = m_j))$ are considered.

$$E(A \mid \bigwedge_{j=0}^{k}(B_j = m_j)) = \sum_i i \cdot P(A = i \mid \bigwedge_{j=0}^{k}(B_j = m_j)) \tag{8}$$

So a prediction is made by looking at the features of a test data point and by checking which row of the conditional probability matrix it belongs to. Then, the expectation of the row is calculated to predict the delay of the train. For example, for the data point in Utrecht and on a weekday the prediction will be 0.451. So the matrix above is a model for predicting delay for the basis schedule. But as every test data point is assigned a row this model only has four possible predictions. Adding more features and increasing sizes of the possible values for the features will increase the size of the conditional probability matrix and thus increase the number of possible predictions. However, one should be careful not to increase the size of the matrix too much as it can lead to overfitting of the data. Therefore, next chapter different we will compare different features against each other to check which features create the best model. The code for creating the conditional probabilities model is written in Python.

4.2 Important features

In this section the performances on predicting delay of the different features are checked. This is done by training models for all the different combinations of features. Then the models make predictions of the data in the validation set and the performance score is calculated. The best features are chosen to be in the final model. The features that are going to be checked are the delay 20 minutes ago (Del), the time, the activity (A) and the location (L). The first is a numerical feature and the remaining are categorical. The time is checked in three different ways: what day (Day) it is, which hour (H) it is and whether it is rush hour, low hour or weekend (RH), as there is no rush hour in the weekend. RH can not be combined with H and Day, but Day and H can be combined together. The feature the model aims to predict is the delay at the current moment, which is also numerical.

Maxima of the numerical features

The first thing that needs to be set is the maxima of the numerical features. Every data point with a value higher than the maximum will get the maximum as value. This is done because high numbers of minutes delay don't occur often and if the model is trained too much on these special cases this could lead to overfitting. The maxima are determined by creating models varying the number of minutes delay until the performance on the validation set stays about the same. The results for the different schedules are seen in the Figures 8 and 9.

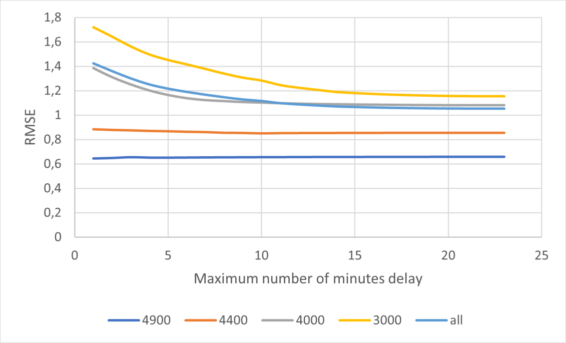

Figure 8: Performance in RMSE per maximum number of minutes delay in the basis schedule.

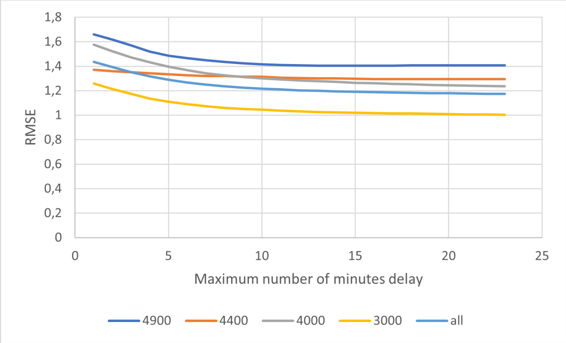

Figure 9: Performance in RMSE per maximum number of minutes delay in the regular schedule.

You can see for the basis schedule that the lines somewhat stop decreasing around 17 minutes delay and for the regular schedule around 20 minutes delay, so these become the maximums for both numerical features. Once the maximum for the to be predicted feature is determined, the models for all the different features can be created. The performances are calculated by taking the root mean squared error (RMSE) of the predictions and the true labels.

Performance of the features

In Table 4 you can see the performances of the models trained on different features compared against each other. They are also compared to the baseline, which is the expectation of the amount of minutes delay as seen in Equation 7 without any conditions. Table 4 only shows the performances of the models trained on one feature, for the performances of all the combinations of features see Table 10 in the Appendix.

As you can see the results of all the combinations of features without the feature Del are very similar to that of the results of the baseline. These features have almost no influence on the number of minutes delay. It could be that they are independent of the number of minutes delay, which is defined as follows:

$$A \text{ and } B \text{ are independent if } P(A \mid B) = P(A) \tag{9}$$

So in order for this to be true for a feature B, for all events $B = m$ and $A = i$ the probability $P(A = i|B = m) = P(A = i)$. The best way to check this is to look at the distributions of $P(A|B = m)$ for every event $B = m$ and compare them to the distribution of $P(A)$. This has been

	Basis schedule					Regular schedule				
	4900	4400	4000	3000	all	4900	4400	4000	3000	all
Baseline	0,673	0,890	1,467	1,783	1,487	1,701	1,406	1,682	1,362	1,526
Delay	0,657	0,855	1,088	1,181	1,066	1,406	1,294	1,251	1,013	1,183
Day	0,668	0,895	1,466	1,788	1,489	1,694	1,404	1,683	1,365	1,526
Hour	0,680	0,893	1,467	1,784	1,488	1,699	1,404	1,684	1,367	1,528
Rush hour	0,673	0,890	1,467	1,785	1,488	1,696	1,405	1,681	1,362	1,525
Activity	0,673	0,890	1,467	1,783	1,487	1,701	1,405	1,683	1,362	1,526
Location	0,672	0,890	1,467	1,782	1,486	1,697	1,406	1,684	1,362	1,526

Table 4: Performance of the different features of the conditional probabilities model by using mean squared error.

done for the 4000 series for every feature. In Figures 10 and 11 the distributions of the feature *Day* are shown per schedule. The pink dotted line is $P(A)$, the distribution of the number of minutes delay without any conditions. So it is the distribution of all the data together. The other lines represent different events. For example, the orange line represents the distribution of the subset $Day = Monday$, where at all the data points the day was Monday. The distributions of the events lie very close to the distribution of the number of minutes delay (the pink dotted line). For the regular schedule, we see that the weekend days have a slightly higher prediction of arriving on time than the weekdays. It is clear that these conditional probabilities are very similar to that of $P(A)$, and thus will have almost no influence on the number of minutes delay.

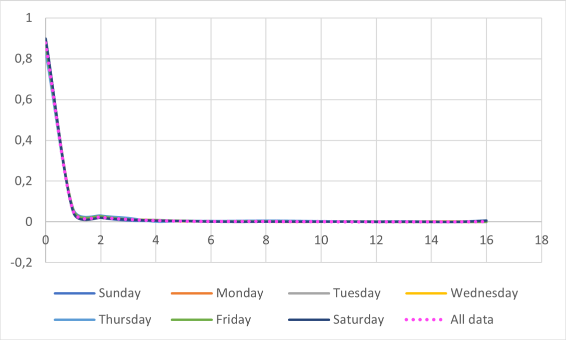

Figure 10: Distribution of feature *Day* in the basis schedule for 4000 series.

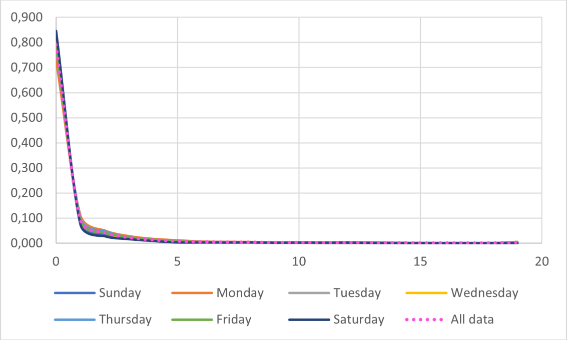

Figure 11: Distribution of feature *Day* in the regular schedule for 4000 series.

In the Figures 12 and 13 we see the distributions of the feature *Del*, the amount of delay 20 minutes before. The distributions of the events are very different from the distribution of the amount of minutes delay, except for the event that the train 20 minutes ago had zero minutes delay. It is clear that the features are dependent on each other. When a train 20 minutes ago has twenty minutes delay, that train will never arrive on time. But when a train 20 minutes ago has four minutes delay, it could drive faster to decrease the delay and so there is a probability it will arrive with no delay. The differences between the two schedules are mostly that in the basis schedule the peaks per distribution are on lower numbers of minutes delay than those of the regular schedule. This means that in the basis schedule it is easier to decrease the delay than in the regular schedule.

In Figures 14 and 15 you can see the distributions of the feature *Hour*. The most notable distributions are the grey line representing 02:00, the yellow line representing the hour 03:00 and the

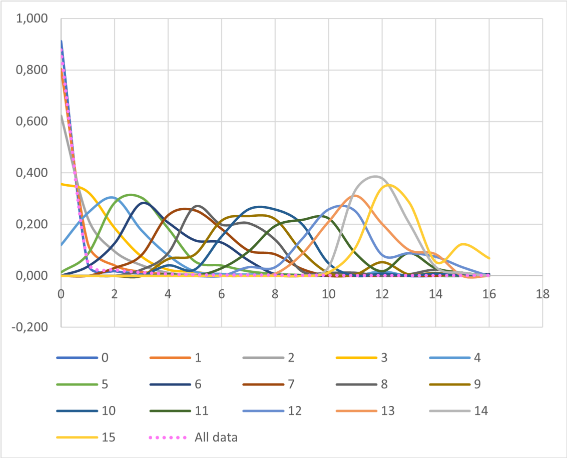

Figure 12: Distribution of feature *Delay* in the basis schedule for 4000 series.

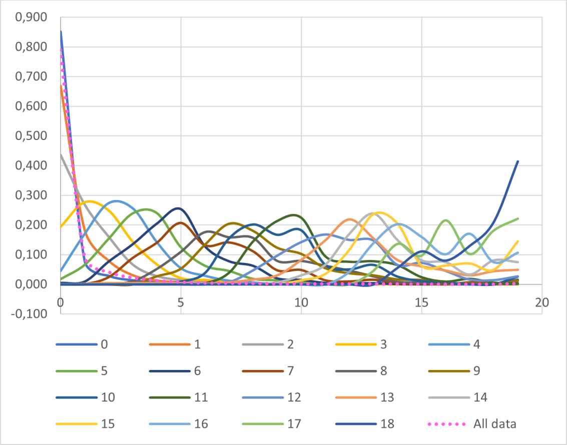

Figure 13: Distribution of feature *Delay* in the regular schedule for 4000 series.

blue straight line representing the hour 04:00. These are interesting as they are hours not included in the daily schedule, so there aren't supposed to be travelling any passenger trains at that time. For the hour 04:00 and in the regular schedule the hour 03:00, there are no data points. So that's why the distribution coincides with the x-axis. For the hour 03:00 in the basis schedule, there are data points. There is one instance of a train having 61 minutes delay at 01:00. The train continued its route at 03:00 and arrived at the remaining stations on his route with a delay of 1 and 0 minutes. Traffic control adjusted the schedule, as the train would never be able to arrive on time. In Chapter 3 you can read why this happens. Because in the data set there is just one train ride at 03:00, there are not enough data points to accurately predict train rides in this hour. For the hour 02:00, the probability of a train having no delay is almost one. Only on Fridays and Saturdays there is a special night train from Amsterdam (Asd) to Heerhugowaard (Hwd). The last train of the series 4000 before this special train has already been scheduled to stop an hour earlier, which shows that this train ride is an exception to the schedule as there are two train rides per hour for every other hour. Because of the late hour, there are fewer trains riding and fewer people are travelling by train. This could be the explanation for the absence of delay. The rest of the hours have distributions similar to that of $P(A)$.

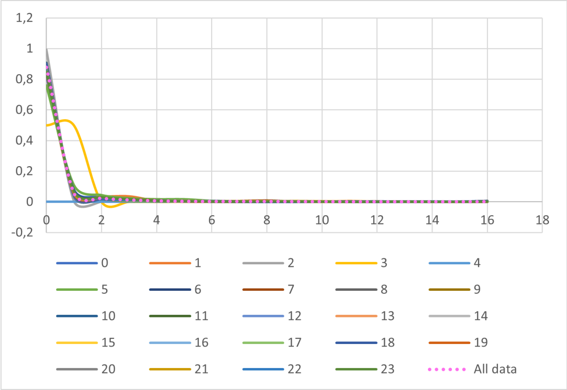

Figure 14: Distribution of feature *Hour* in the basis schedule for 4000 series.

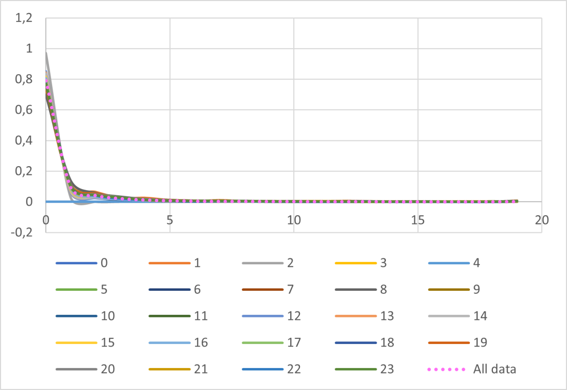

Figure 15: Distribution of feature *Hour* in the regular schedule for 4000 series.

In Figures 16 and 17 you can see the distributions of the feature location. As seen before the schedule switches from day to day. Some days there is a train riding at 02:00 and in the same way some days there is a train riding to Utrecht. This happened four times in the regular schedule, so for every location on that route there are only four data points. As these locations are not in the normal schedule and have too few data points to make an accurate distribution, they should be left out. Then there are a few locations that are not included in every train ride. The series 4000 with direction E travels from Rotterdam (Rtd) to Uitgeest (Uitg) and with direction O travels the other way around. But at the beginning and at the end of the day the 4000 series stops at stations it doesn't stop at the rest of the day. This is because at night trains are held at a storage yard. There are a couple of stations with a storage yard and at night every train in the Netherlands should be held somewhere. This means that at the beginning of the day a train starts at a storage yard and should travel to its route and at the end of the day it is driven back to a storage yard. This causes there to be stations that are only stopped at once or twice a day, and always at the end or the beginning, for example Alkmaar (Amr) and Heerhugowaard (Hwd). The rest of the distributions are similar to that of the distribution of the amount of minutes delay.

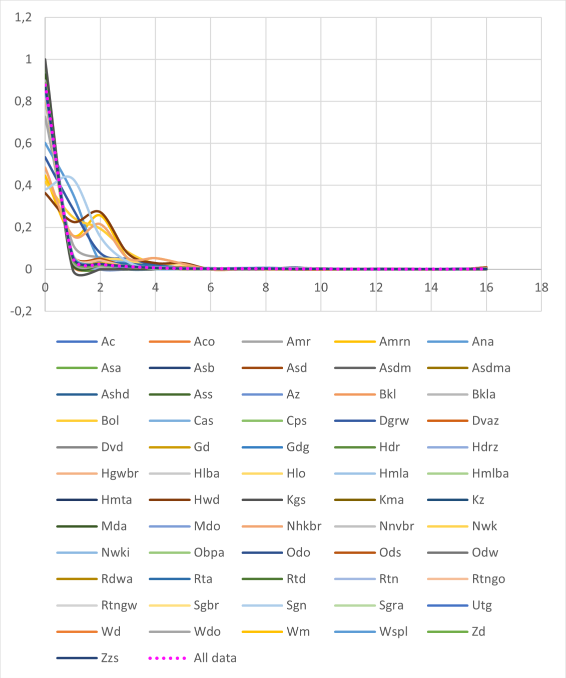

Figure 16: Distribution of feature *Location* in the basis schedule for 4000 series.

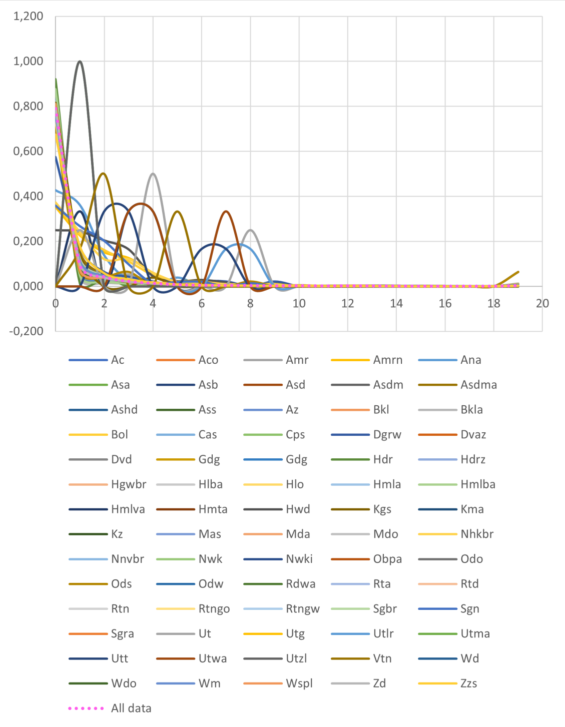

Figure 17: Distribution of feature *Location* in the regular schedule for 4000 series.

In Figures 22 and 23 in the Appendix you can see the distributions of the feature *Activity* and in Figures 24 and 25 the distribution of the feature Rush hour. As you can see these are all very similar to the distribution of $P(A)$, so they have almost no influence on the amount of delay.

In Table 10 in the Appendix the performances of the combinations of features are checked. The best combinations are the ones combined with the feature *Del*. The more features there are in the condition, the worse the performance of the model. The cause of this can be explained by overfitting. The more features in the condition, the more unique events there are. For each event there are fewer data points, which makes it hard to get an accurate prediction for this event. In some cases, there is even no data as it has not happened in the data set. Therefore, these models perform well on their trained data set, but not on other data sets. In the regular schedule the best models are *Del*, *Del* ∧ *RH*, *Del* ∧ *Act* and *Del* ∧ *RH* ∧ *Act*. This is because for each of these models for low values of the feature *Del* the distributions are the same, which means that, for example, the event one minute delay twenty minutes ago during rush hour arriving at a station has roughly the same distribution as the event one minute delay twenty minutes ago during the weekend. As high values of delay occur less, there are a lot of unique events and some events do not occur in the data set. Therefore the best model for the regular schedule is the model with the feature *Del* as a condition. For the basis schedule the performance of the model with the feature *Del* is the best, with a larger difference in performance to the other models than the regular schedule.

Final models for the different series

In Figure 18 the final models for the series 4000 are compared to each other and the current method. The models follow somewhat the same curve, but the basis schedule predicts lower than the regular schedule. The current method lays between the two models. The conditional probabilities models both predict that the delay will decrease, except for a couple of exceptions. Both the models predict a little increase when the number of minutes delay twenty minutes ago is zero, but the regular model also predicts an increase at the numbers 12 and 14. The basis model is below the current method except for two points, which means that it overall predicts a decrease of more than one minute. The regular model is for a couple of points slightly below the current method and for the other points above, which means that it overall predicts a decrease of less than one minute. The difference between the regular model and the basis model increases when the number of minutes delay 20 minutes ago increases. So at large delays the basis model is better at predicting cases where the delay decreases and the regular model at cases where the delay stays the same or increases.

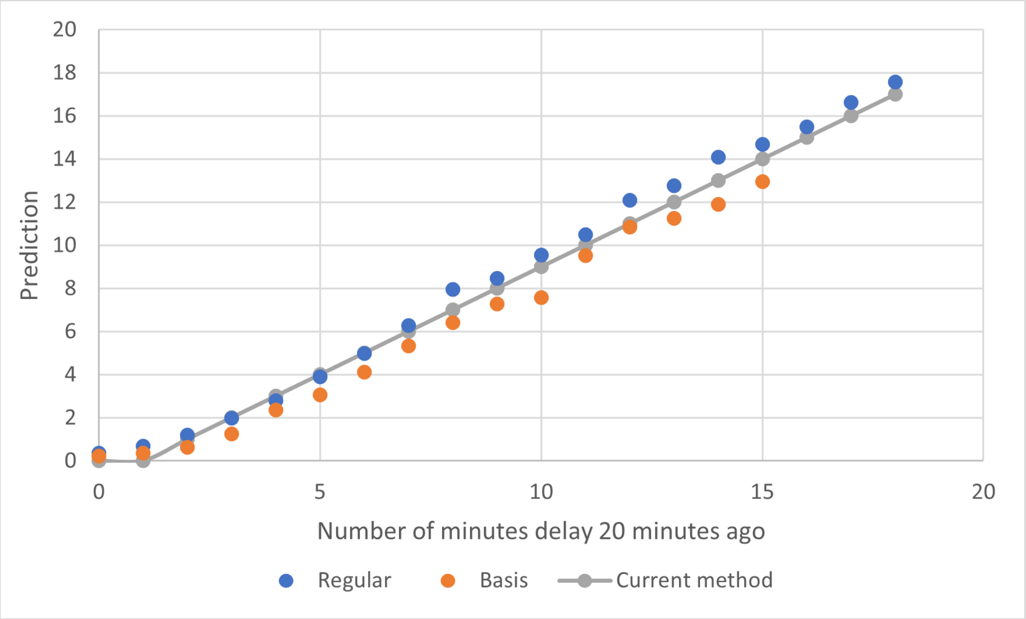

Figure 18: Conditional probabilities models for the 4000 series and current method.

The curves of the models for the series 4900 and 3000 are seen in the Appendix in Figures 26 and 27. The 3000 series has a similar basis model and regular model as the 4000 series. In the 4900 series the basis model rises above the current method when the delay 20 minutes ago is six to eight minutes and 15 minutes. The regular model for the 4900 is below the current method more often and lower than the regular model for the 4000 series.

The models for the 4400 series are drastically different from the models for the other series. This can be seen in Figure 19. The regular model is below the current method for delay 20 minutes ago greater than two minutes and keeps a distance of an average of four minutes from the current method. The basis model is even further from the current method as it is below the regular model and makes a giant leap from 11 minutes delay to 13 minutes delay. At 11 minutes delay 20 minutes ago the prediction is around four minutes delay, at 12 there are no data points and at 13 the prediction is 15 minutes delay. After that, the prediction becomes around 5 minutes delay again. So the prediction of delay is depended on what train series it is.

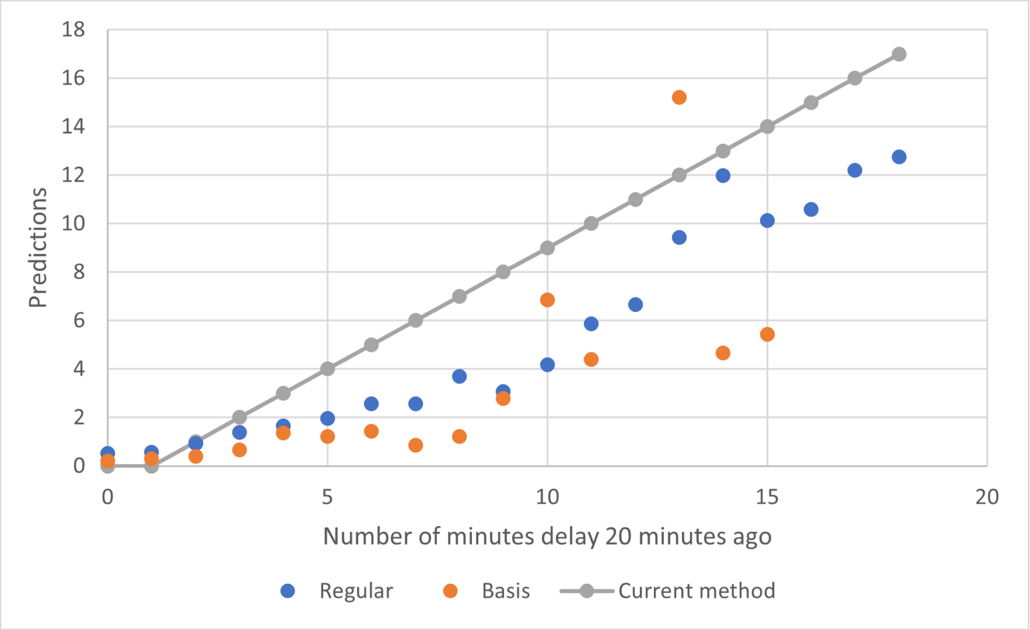

Figure 19: Conditional probabilities model for series 4400.

The 4900 series and the 4400 series have significantly fewer data points than the other series. Therefore there is less data on jumps in delay and the chances of creating a wrong prediction become higher. Therefore an attempt was made to create trend lines in Excel for these models. These lines made predictions for the validation set and no line was better at predicting the delay than the original models. So the models were kept the way they are.

4.3 Results

In this paragraph, the conditional probabilities models trained on the regular schedule (regular conditional model) and the conditional probabilities models trained on the data from the basis schedule (basis conditional model) are tested on the two test sets. The performance of the models on the different train series and schedules are given in table 5. Again the RMSE (see Equation 2) is used to determine the performance of the models. As seen in Section 2.4: the lower the performance score, the better the performance.

In almost all the cases the models are better at predicting the test set of their own schedule than the other model. Only at the 4900 series is the regular conditional model with a very small difference

Test set	Training set	4900	4400	4000	3000	All series
Basis	Basis	1,9783	0,8589	0,5911	0,7064	0,9192
Basis	Regular	1,9733	0,9077	0,6480	0,7161	0,9424
Regular	Basis	1,0761	1,3757	1,2871	1,2839	1,2796
Regular	Regular	1,0470	1,3204	1,2633	1,2804	1,2587

Table 5: Performance of conditional probabilities models on different test sets.

better than the basis conditional model on the basis test set. In fact, the differences between the performances of the models on the same test set are small for all series, with 0.0569 in RMSE being the largest difference.

The basis test set is easier to predict, as the performance scores on the basic test set from both models is lower than the performance scores on the regular test set. Only at the 4900 series is the performance score a lot higher on the basis test set than the regular test set. This is also seen in the performances of the Neural Network models and is explained in 3.3.

5 Comparison of the models

In this section, the different methods of creating models are compared against each other. The qualitative aspects of each method in respect to NS's wishes are discussed and a closer look is given to the differences in performance scores.

5.1 Results

In Section 3.3 we saw the results of the performance score of the models created using Neural Networks and in Section 4.3 the results of the performance score of the models created using conditional probabilities. The combination of these results is seen in Table 6. They are ordered from lowest overall performance score on the basis test set to highest and similarly ordered for the regular test set.

Model	Training set	Test set	4900	4400	4000	3000	All series
Conditional probabilities	Basis	Basis	1,9783	0,8589	0,5911	0,7064	0,9192
Neural Networks	Basis	Basis	1,9741	0,9406	0,6280	0,6796	0,9326
Conditional probabilities	Regular	Basis	1,9733	0,9077	0,6480	0,7161	0,9424
Neural Networks	Regular	Basis	1,9908	0,8802	0,7189	0,7128	0,9617
Conditional probabilities	Regular	Regular	1,0470	1,3204	1,2633	1,2804	1,2587
Neural Networks	Regular	Regular	1,0347	1,2959	1,2884	1,2697	1,2593
Conditional probabilities	Basis	Regular	1,0761	1,3757	1,2871	1,2839	1,2796
Neural Networks	Basis	Regular	1,1299	1,3973	1,3622	1,3213	1,3301

Table 6: Performance of the models on different test sets.

The model that performs the best on the basis test set is the conditional probabilities model trained on the basis schedule. For the regular test set the best performance overall is by the conditional probabilities model trained on the regular schedule. The models that score second are the Neural Networks models trained on the respective schedules. So the models trained on a schedule do outperform the other models on that schedule's test set. For some train series the Neural Networks models perform better than the conditional probabilities models. The differences between the performance score of the methods are very small, for the regular test set even around 0.0006 in RMSE. Therefore the conditional probabilities method is not significantly better than the method using Neural Networks.

The difference in performance score for the different training sets is slightly larger. Especially in the case of the regular schedule is there a larger gap. The difference is still very small, less than 0.1 in RMSE for all the models. In the next paragraph, these differences are better visualised.

5.2 Difference in performance score

In the paragraph above the performances of the models on the different test sets are given using the RMSE (Equation 2). The results seem very close to each other, but what does a difference of 0.1 in RMSE mean for these models? For example, this difference could have been caused by one model having a very bad prediction for one of the data points and the other model predicting that data point accurately or a lot of close predictions but the other model a little bit closer. In this paragraph, a visualisation of the differences in RMSE is given.

It is hard to find out what the cause of the difference in RMSE is. We now take a closer look at what it means in RMSE when two models perform the same on 99% of the test data and for 1% of the test data one model predicts one minute closer to the true label. Suppose the first model w predicts accurately for all the test data x and the second model v accurately for 99% and with a difference of one minute for 1%. Then the difference in RMSE becomes the following, where w_d and v_d are the predictions of the respective models on data point d.

$$\begin{aligned}\Delta\,\text{RMSE} &= \sqrt{\frac{\sum_{d=1}^{D}(v_d - x_d)^2}{D}} - \sqrt{\frac{\sum_{d=1}^{D}(w_d - x_d)^2}{D}} \\ &= \sqrt{\frac{\sum_{d=1}^{0.01\cdot D}(1-0)^2 + \sum_{d=0.01\cdot D+1}^{D} 0^2}{D}} - \sqrt{\frac{\sum_{d=1}^{D} 0^2}{D}} \\ &= \sqrt{\frac{0.01\cdot D}{D}} - 0 = 0.1\end{aligned} \tag{10}$$

Now suppose the first model predicts accurately for just 1% of the data points, and for 99% with a difference of two minutes. And suppose the second model predicts for 99% of the test data with a difference of two minutes and for 1% of the data with a difference of 1 minute. The difference in RMSE then becomes the following.

$$\Delta\,\text{RMSE}^* = \sqrt{\frac{0.99\cdot D\cdot 2^2 + 0.01\cdot D\cdot 1^2}{D}} - \sqrt{\frac{0.99\cdot D\cdot 2^2}{D}} = \sqrt{0.99\cdot 4 + 0.01} - \sqrt{0.99}\cdot 2 \approx 0.002511 \tag{11}$$

So $\Delta RMSE$ is 40 times larger than $\Delta RMSE^*$. This is caused by the curve of the square root. In Figure 20 you can see how the square root $y = \sqrt{x}$ behaves. When a small change δ is added to the input x it has a lot more influence on the output y when x is smaller than 1. This is because the slope of the function gets less steep when x gets larger. This means that when the error of the models is already high for the 99% of the test data, a difference of 1 minute on 1% of the data has a lower impact than when the error of the models for the 99% of test data is low.

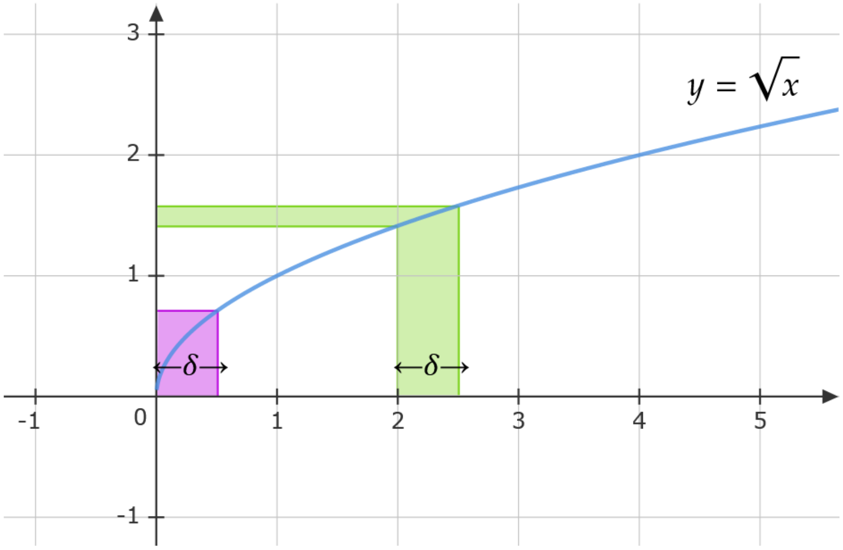

Figure 20: Curve of square root.

In the same way, you can see that the difference is also dependent on the error of the 1% of data. In the examples above the errors were 0 for the first model and 1 for the second model, but suppose they were 1 and 2 respectively. Then when for 99% of the data the models predicted with a difference of 1 minute from the true label, this difference becomes approximately 0, 014889. And when they are 2 and 3 the difference is approximately 0, 024341.

Therefore a difference in RMSE cannot be translated to adding 1 minute of difference between prediction and true label for a single percentage of test data, as it is clear that the RMSE is dependent on the height of the error of the models on the test data. So for every two models, the difference in RMSE is different in terms of this percentage. Suppose we have a model w with the performance score a. Then we can translate this model to the model $\hat{w}$ that always predicts a away from the true label, so $|\hat{w}_d - x_d| = a$ for all data points d. Then the performance score becomes:

$$RMSE = \sqrt{\frac{\sum_{d=1}^{D} a^2}{D}} = \sqrt{\frac{D \cdot a^2}{D}} = a \tag{12}$$

So these two models have the same performance score. Now let's take a look at the model $\hat{v}$ that performs similar for $(1-p) \cdot D$ data points to $\hat{w}$, so $|\hat{v}_d - x_d| = a$, and for $p \cdot D$ data points it performs 1 minute worse than $\hat{w}$, so $|\hat{v}_d - x_d| = a + 1$. Then the performance score is the following.

$$\begin{aligned} RMSE &= \sqrt{\frac{\sum_{d=1}^{pD}(a+1)^2 + \sum_{d=pD+1}^{D} a^2}{D}} \\ &= \sqrt{\frac{p \cdot D \cdot (a+1)^2 + (1-p) \cdot D \cdot a^2}{D}} \\ &= \sqrt{p \cdot (a+1)^2 + (1-p) \cdot a^2} \geq a \end{aligned} \tag{13}$$

If we have the performance score of model v in RMSE, then we can compute the percentage p such that the model v has the same performance score as the model $\hat{v}$. Then w and v have been translated to $\hat{w}$ and $\hat{v}$, so we can now compare them in terms of percentage of data points where w performs one minute better than v. Every performance of the models is compared this way in Tables 7 and 8 for the basis test data and the regular test data.

w v	Basis Conditional	Basis Neural	Regular Conditional	Regular Neural
Basis Conditional	0%	■	■	■
Basis Neural	0.874%	0%	■	■
Regular Conditional	1.522 %	0.641%	0 %	■
Regular Neural	2.816%	1.924%	1.274%	0%

Table 7: Percentage of data points where model w performs one minute better than model v on the basis test data.

w v	Regular Conditional	Regular Neural	Basis Conditional	Basis Neural
Regular Conditional	0%	■	■	■
Regular Neural	0.043%	0%	■	■
Basis Conditional	1.508 %	1.465%	0 %	■
Basis Neural	5.255%	5.210%	3.703%	0%

Table 8: Percentage of data points where model w performs one minute better than model v on the regular test data.

The conditional probabilities models have the same difference on the basis test set and the regular test set, only the basis model performs better on the first and the regular model on the second. The

Neural Networks models have a larger difference on the regular test set than on the basis test set. This could be explained by the basis schedule being easier to predict. So the models created using Neural Networks are tailored towards predicting the schedule they are trained on, where the conditional probabilities models perform well on both schedules.

The percentages between the best performing conditional probabilities model and the best performing Neural Networks model on each test set are very low. Therefore the one method does not significantly outperform the other method.

5.3 Comparison

We have already seen that the performance scores of the models created using Neural Networks and the models created using conditional probabilities are very similar. In this section, the qualitative aspects of the different methods are compared against each other.

An example of a jump in delay and the predictions of these data points from the different models is seen in Figure 21. The green line shows what actually happened and the other lines show what the predictions are of the models.

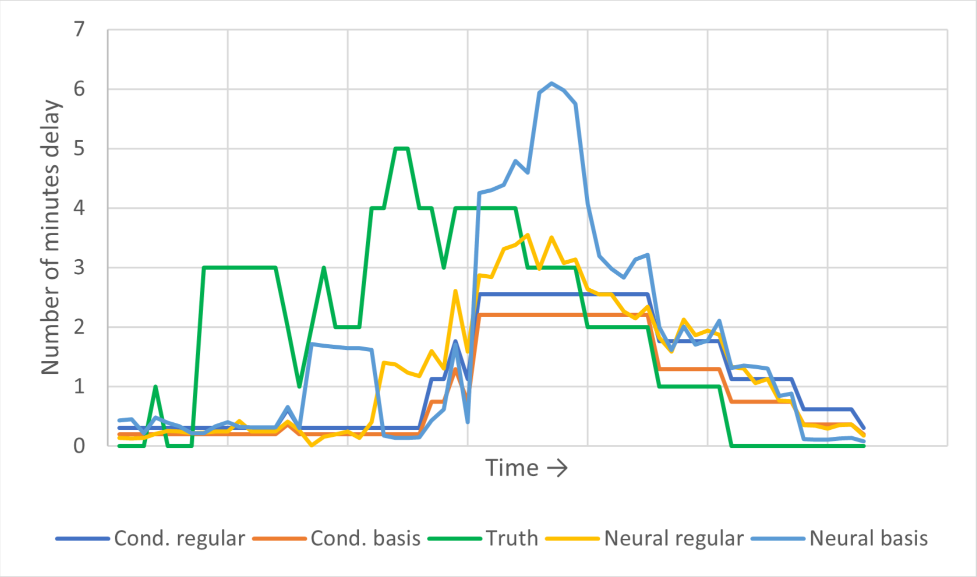

Figure 21: Example of a progression of a jump in delay in the regular test set for the series 3000 and the predictions of the models.

You can see the predictions of the models increase later than the true label, as they know of this delay 20 minutes later. The predictions of the conditional probabilities models are only based on the delay 20 minutes ago, so there are a lot of flat lines. This makes it easy to see why the predictions are made. The model created using Neural Networks trained on the regular schedule (yellow line) moves roughly similar to the conditional probabilities models, but has no flat lines as it has a lot more possible outputs. The model created using Neural Networks trained on the basis schedule (light blue line) predicts a large increase in delay that did not happen. It is hard to understand why the model made this decision, as you can not look into the model. This makes it hard to trust the outputs of the model. The conditional probabilities model is based on the frequencies of situations occurring and is a function of the delay 20 minutes ago. Therefore the predictions are better to understand and the accuracy of the conditional probabilities models become more believable than the Neural Network models.

Schedule	Series	Test	Predicted jumps	Correct jumps	Actual number of jumps
Basis	3000	Basis	0	0	199
Regular	3000	Basis	0	0	
Basis	3000	Regular	25	2	387
Regular	3000	Regular	4	0	
Basis	4000	Basis	0	0	134
Regular	4000	Basis	28	3	
Basis	4000	Regular	93	6	491
Regular	4000	Regular	0	0	

Table 9: Number of jumps predicted by the Neural Networks models in comparison to the actual jumps.

As told in the introduction, the NS wishes to have a predictor where the number of wrongfully predicted increases in delay is low. Consumers find wrongfully predicted increases in delay worse than wrongfully predicted decreases. Table 9 shows the number of jumps the Neural Networks models predicted on the test sets, where a jump is an increase of greater of equal to 2 minutes delay. The conditional probabilities models never predict an increase in delay larger than or equal to two minutes, except only for the 4400 basis model when the number of minutes delay 20 minutes ago is 12. The Neural Network models predicted increases in delay more often on the test sets, but not for the series 4400 and 4900. Therefore these two series are not included in the table. The ratio of right predicted jumps to wrong predicted jumps is less than 1/8 for each of the models. The ratio of right predicted jumps to actual jumps is even lower, less than 1/44. So when the Neural Network model predicts an increase in delay it more often predicts wrong than right and it misses almost all the increases.

6 Discussion

This research focused on predicting the delay of trains twenty minutes into the future for two different operating schedules. These schedules were the regular schedule and the basis schedule. The latter took place at the start of the COVID-19 pandemic and is a reduced version of the regular schedule. Two methods for creating models were studied. The first method was the machine learning technique Neural Networks and the second the classic statistical technique using conditional probabilities. Both methods were trained on the two different schedules, resulting in four different models. All the models were tested on the test set from the basis schedule and the test set from the regular schedule. Results were compared to each other to see which model performed the best on which test set.

When creating the conditional probabilities model, the importance of the different features was discussed. All the features did not have much influence on the delay twenty minutes into the future, except for the current delay. Even when looking at all the possible combinations of features, the best performing model was the model that only looked at the current delay. This was the case for both the operating schedules. The final conditional probabilities models only looked at what train series the train is and what the current delay is.

Results show that the models trained on their respective schedule performed the best on that schedule's test set. This is an expected result, as they were trained on that schedule. However, the differences between the performances of the differently trained models was smaller than expected. A large difference in schedule did not cause the models to significantly outperform the other models. This shows that when there is a small difference in operating schedule, it is not necessary to train the models again on an updated data set.

The method that performed the best overall was the conditional probabilities method, but the difference with the performance of the models created using Neural Networks was small. It is even smaller than the differences between the schedules. Therefore, it can not be said that the conditional probabilities method significantly outperforms the Neural Networks method.

Neural Networks are used to find connections between features that are hard for humans to see. Therefore a model created using a Neural Network is similar to a 'black box'. You input data and predictions come out. The mathematical equations are easily computed, but to understand why the model created these outputs is difficult. It can suddenly predict a jump in delay, but the reasons for this decision are unknown. The conditional probabilities method on the other hand is a more intuitive method. It is based on the frequencies of situations occurring and it is known on what features it makes its prediction. This makes it easier to trust the model.

The NS wants a model where the chance of a wrongfully predicted increase in delay is low. For customer satisfaction it is better to wrongfully predict a decrease in delay. Only the Neural Networks method predicted increases in delay on the test sets and they almost never predicted them correctly.

7 Further research and limitations

The results of this thesis are based on the features of the models. There could be features, that when added, will result in a better predictor for each of the methods. More research could be done in adding features. A feature that has not been looked at, for example, is the number of stops between the data point twenty minutes ago and the current data point. Eva Lehkà attempted to add more infrastructure to the Neural Network model, but this turned out to be harder than expected. However, the creation of the features is limited by the data provided. Also the models were only trained and tested on the data of the train series 4900, 4400, 4000 and 3000. In this thesis we already saw differences between the models trained on these train series, so there could be interesting results for train series that have not been looked at.

This thesis only concerns short term delay prediction, as only the delay twenty minutes ago is looked at. This is the most influential feature, so the model could be expanded by adding features concerning the delay earlier and/or later. In practice, the predictions of delay are not based on the delay twenty minutes ago, but on the delay of the latest known measuring point. Adding the features concerning delay at different times in the past therefore is more practical. It also could show how a jump in delay progresses in time. For example, when a jump in delay just happened then the probability of an increase in delay could be higher than when the previous delays show a decrease in delay. Researching this could result in a better predictor.

Statistical methods and especially Neural Networks are known to be good at predicting the bulk of the data. In the regular schedule for almost 80% of the data points the number of minutes delay is zero. Saying the delay is always zero is already a good predictor, as it is correct around 80% of the time. So when researching the delays of trains, one tries to predict the other 20% of the data. These are the outliers and therefore hard to predict. Neural Networks are better at predicting these outliers than other methods, but it still more often predicts them wrong than right. When taking the wishes of the Nederlandse Spoorwegen into account, the conditional probabilities method is better. So my advice is to stop investing time in the research of machine learning techniques when predicting the delay of trains.

8 Conclusion

The most important feature for predicting the delay of trains is the number of minutes delay the train had twenty minutes ago. This is true for both the basis and the regular schedule.

The models trained on the training set containing the regular schedule perform better on the test set of the regular schedule than the models trained on the basis schedule and vice versa. However, the difference in performance score is only significant for the Neural Network models on the regular test set. Therefore the difference in creating a model for predicting delay for the basis schedule and the regular schedule depends on the method.

The conditional probabilities method performs similar to the Neural Network method on the test sets. But the conditional probabilities method is more intuitive than the Neural Network method. Moreover, the Neural Network models wrongfully predict increases in delay significantly more than the conditional probabilities method, which does not comply with the wishes of the NS.

Based on the findings in this thesis, I would suggest the Nederlandse Spoorwegen stops researching Neural Networks for predicting the delay of passenger trains and propose to take a further look at classic statistical methods.

References

[1] Leonieke van den Bulk (2018). *Predicting Short Term Train Delays in the Dutch Rail Network*, Radboud University Nijmegen.

[2] Eva Lehká (2019). *Short Term Delay Prediction in Passenger Railways*, Delft University of Technology.

[3] Transport plan NS 2020, Nederlandse Spoorwegen, 2020.

[4] Annual report NS 2019, Nederlandse Spoorwegen, 2020.

[5] L.P.A. van der Breggen (2015), *Voorspellingsmodel voor treinvertragingen*, Vrije Universiteit Amsterdam.

[6] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[7] Michael A. Nielson. *Neural Networks and Deep Learning*, Determination Press, 2015.

A Appendix

	Basis schedule					Regular schedule				
	4900	4400	4000	3000	all	4900	4400	4000	3000	all
Baseline	0,673	0,890	1,467	1,783	1,487	1,701	1,406	1,682	1,362	1,526
Delay	0,657	0,855	1,088	1,181	1,066	1,406	1,294	1,251	1,013	1,183
Day	0,668	0,895	1,466	1,788	1,489	1,694	1,404	1,683	1,365	1,526
Hour	0,680	0,893	1,467	1,784	1,488	1,699	1,404	1,684	1,367	1,528
Rush hour	0,673	0,890	1,467	1,785	1,488	1,696	1,405	1,681	1,362	1,525
Activity	0,673	0,890	1,467	1,783	1,487	1,701	1,405	1,683	1,362	1,526
Location	0,672	0,890	1,467	1,782	1,486	1,697	1,406	1,684	1,362	1,526
Del ∧ Day	0,658	0,886	1,162	1,352	1,173	1,468	1,335	1,263	1019	1,203
Del ∧ H	0,663	0,898	1,190	1,572	1,285	1,472	1,330	1,257	1,063	1,215
Del ∧ RH	0,660	0,931	1,092	1,180	1,076	1,405	1,294	1,253	1,013	1,183
Del ∧ A	0,662	0,866	1,091	1,190	1,073	1,406	1,285	1,252	1,013	1,182
Del ∧ L	0,661	0,872	1,121	1,418	1,186	1,452	1,300	1,265	1,019	1,196
Day ∧ H	0,718	0,914	1,480	1,793	1,500	1,706	1,422	1,702	1,379	1,542
Day ∧ A	0,667	0,895	1,467	1,788	1,489	1,693	1,404	1,683	1,365	1,526
Day ∧ L	0,664	0,897	1,467	1,789	1,490	1,691	1,405	1,685	1,364	1,526
H ∧ A	0,680	0,893	1,467	1,783	1,487	1,699	1,404	1,684	1,368	1,528
H ∧ L	0,686	0,900	1,466	1,784	1,488	1,698	1,404	1,687	1,368	1,529
RH ∧ A	0,673	0,890	1,468	1,785	1,488	1,696	1,406	1,682	1,362	1,525
RH ∧ L	0,674	0,890	1,468	1,784	1,487	1,694	1,406	1,683	1,361	1,525
A ∧ L	0,689	0,882	1,468	1,781	1,486	1,698	1,406	1,685	1,361	1,526
Del ∧ Day ∧ H	0,695	0,919	1,444	1,794	1,486	1,667	1,486	1,498	1,141	1,381
Del ∧ Day ∧ A	0,659	0,900	1,178	1,514	1,253	1,489	1,339	1,274	1,017	1,209
Del ∧ Day ∧ L	0,668	0,901	1,391	1,765	1,450	1,577	1,362	1,441	1,133	1,328
Del ∧ H ∧ A	0,665	0,899	1,250	1,600	1,320	1,569	1,380	1,276	1,075	1,245
Del ∧ H ∧ L	0,665	0,905	1,417	1,792	1,472	1,618	1,413	1,537	1,241	1,416
Del ∧ RH ∧ A	0,660	0,903	1,092	1,242	1,099	1,414	1,286	1,253	1,013	1,183
Del ∧ RH ∧ L	0,656	0,895	1,189	1,621	1,306	1,502	1,305	1,279	1,039	1,215
Del ∧ A ∧ L	0,678	0,869	1,133	1,433	1,197	1,453	1,297	1,272	1,017	1,198
Day ∧ H ∧ A	0,719	0,920	1,481	1,794	1,501	1,708	1,422	1,703	1,382	1,544
Day ∧ H ∧ L	0,742	0,947	1,513	1,807	1,522	1,713	1,423	1,720	1,399	1,558
Day ∧ A ∧ L	0,666	0,897	1,467	1,788	1,499	1,700	1,404	1,685	1,364	1,527
H ∧ A ∧ L	0,700	0,900	1,466	1,784	1,489	1,697	1,403	1,687	1,369	1,529
RH ∧ A ∧ L	0,676	0,890	1,469	1,783	1,488	1,693	1,407	1,682	1,361	1,524
Del ∧ Day ∧ H ∧ A	0,692	0,928	1,466	1,807	1,501	1,703	1,492	1,600	1,209	1,450
Del ∧ Day ∧ H ∧ L	0,712	0,958	1,509	1,823	1,528	1,731	1,500	1,704	1,361	1,549
Del ∧ Day ∧ A ∧ L	0,671	0,901	1,396	1,768	1,453	1,588	1,363	1,446	1,142	1,334
Del ∧ H ∧ A ∧ L	0,680	0,906	1,418	1,792	1,474	1,619	1,416	1,543	1,250	1,422
Del ∧ RH ∧ A ∧ L	0,749	0,866	1,203	1,635	1,322	1,506	1,303	1,285	1,041	1,218
Day ∧ H ∧ A ∧ L	0,745	0,951	1,514	1,806	1,524	1,716	1,424	1,721	1,404	1,560
Del ∧ Day ∧ H ∧ A ∧ L	0,715	0,963	1,510	1,823	1,530	1,734	1,483	1,704	1,368	1,550

Table 10: Performance of different combinations of features of conditional probabilities model by using mean squared error.

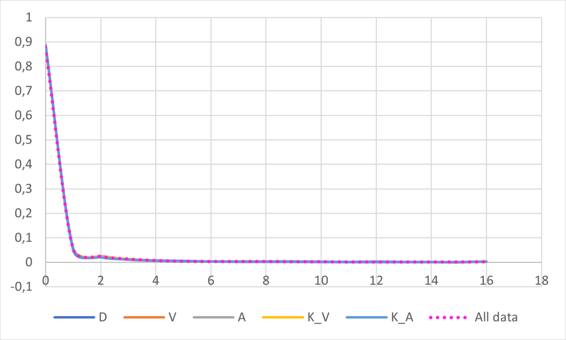

Figure 22: Distribution of feature *Activity* in the basis schedule for 4000 series.

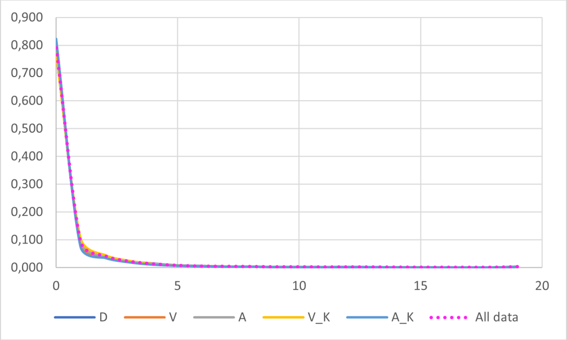

Figure 23: Distribution of feature *Activity* in the regular schedule for 4000 series.

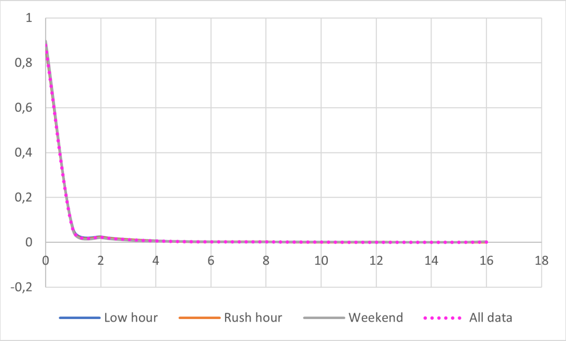

Figure 24: Distribution of feature *Rush hour* in the basis schedule for 4000 series.

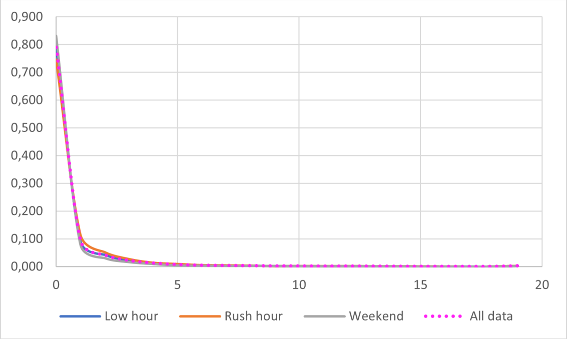

Figure 25: Distribution of feature *Rush hour* in the regular schedule for 4000 series.

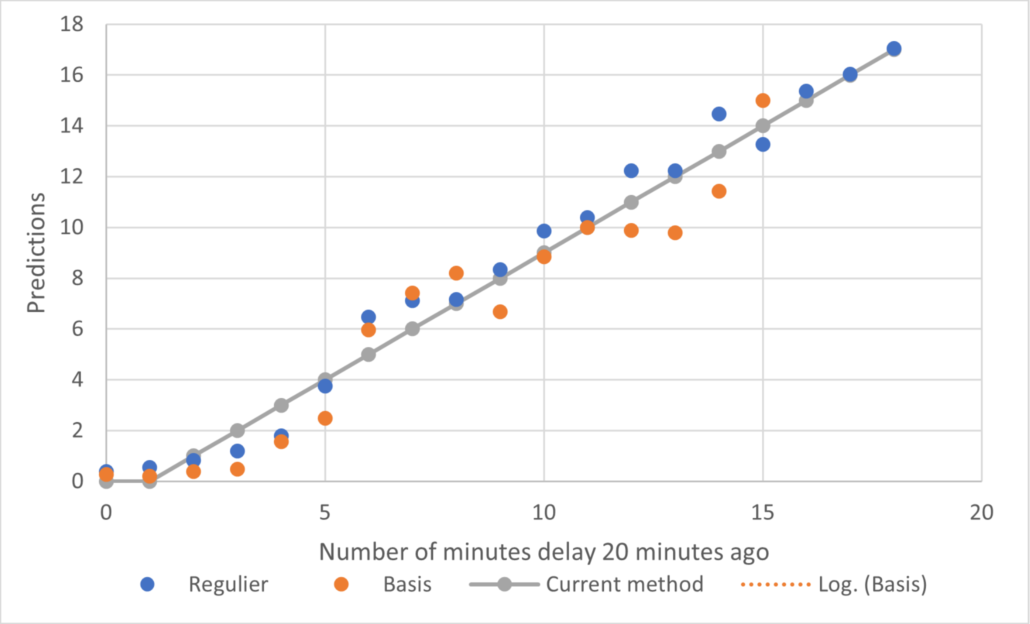

Figure 26: Conditional probabilities model for series 4900.

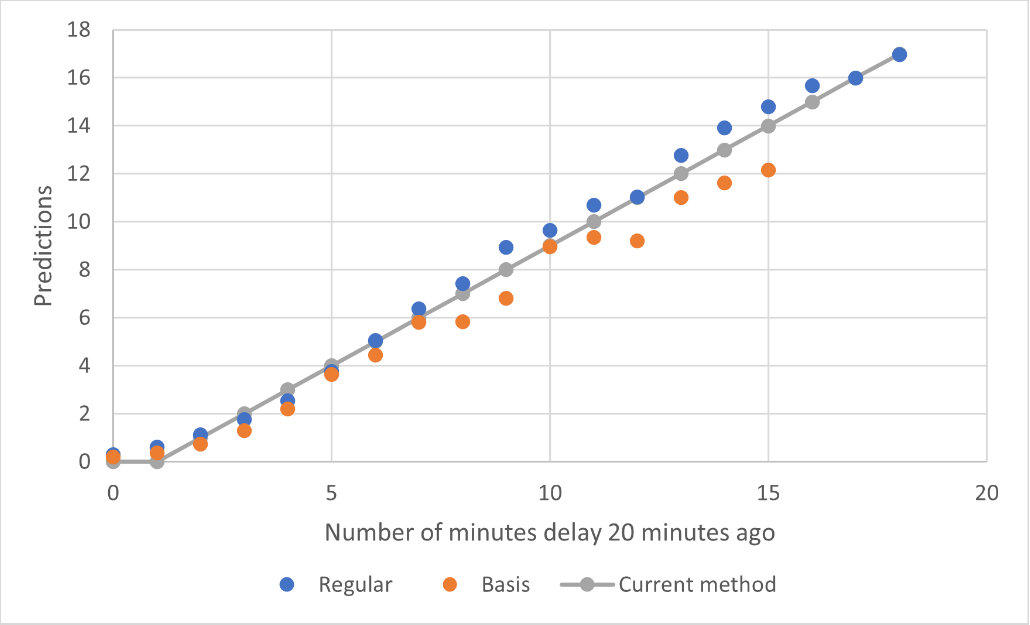

Figure 27: Conditional probabilities model for series 3000.