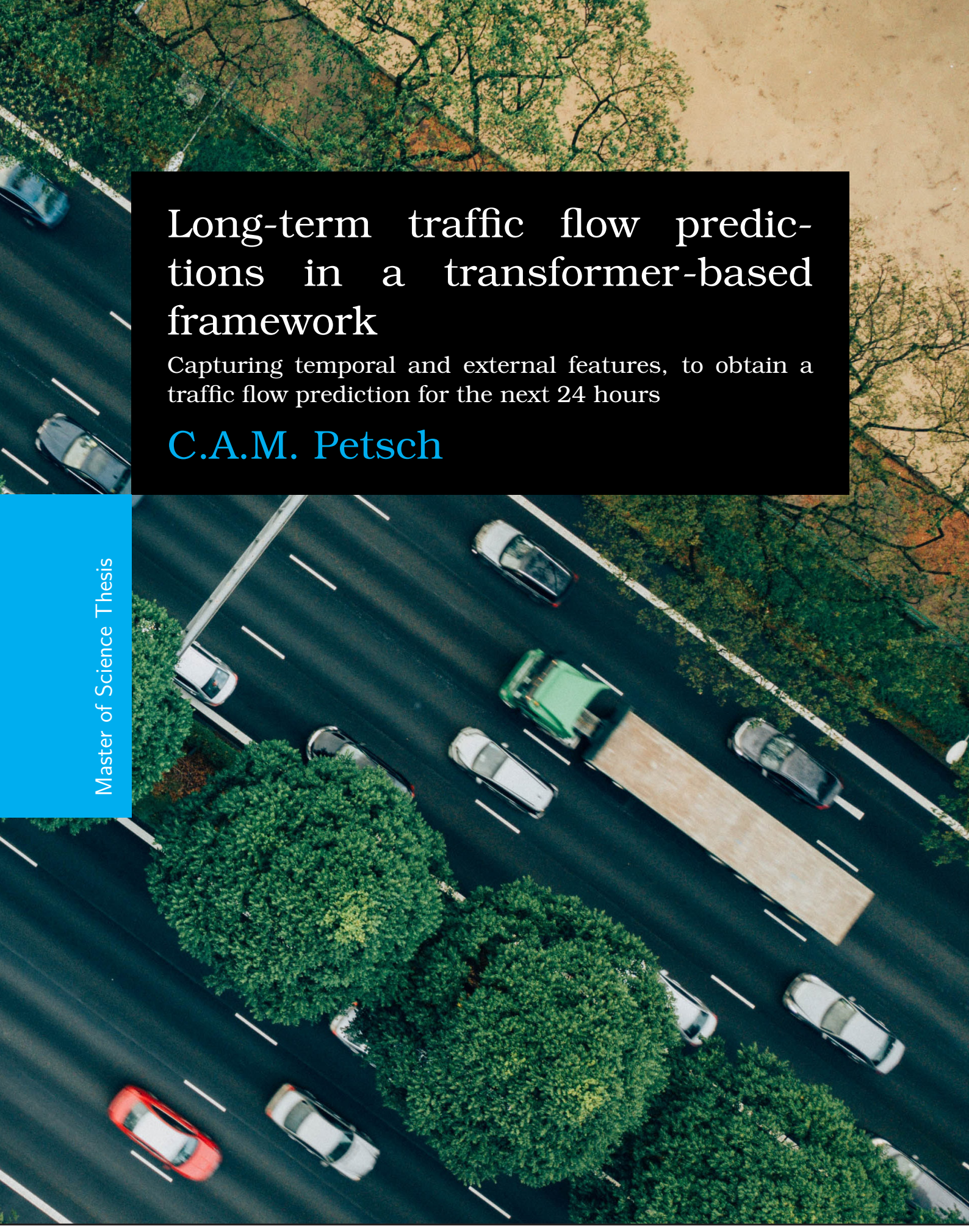


Long-term traffic flow predictions in a transformer-based framework

Capturing temporal and external features, to obtain a traffic flow prediction for the next 24 hours

C.A.M. Petsch

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Long-term traffic flow predictions in a transformer-based framework

Capturing temporal and external features, to obtain a traffic flow prediction for the next 24 hours

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

C.A.M. Petsch

February 16, 2022

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by Siemens Mobility b.v. Their cooperation is hereby gratefully acknowledged.

Abstract

Traffic jams, overcrowded public transport, and limited parking availability are all difficulties that people face on a daily basis. To cope with the increasing pressure on the transportation network, more efficient use of the multi-modal transportation network should be made by distributing travelers over different transportation modes. This requires insights into traffic behavior, highlighting the necessity of a traffic flow prediction model. This research focuses on a fraction of the entire transportation network, by making a long-term traffic flow prediction of vehicles on arterial roads, based on historical data. In this context, long-term is defined as 24 hours ahead.

Traffic flow prediction is a challenging topic due to the highly nonlinear temporal features, especially on longer prediction horizons. In addition, external features may also have a significant impact on traffic behavior. Models utilized to capture temporal features are traditionally based on recurrence. However, due to the recurrent structure, these models are, computationally expensive, time-invariant, and encounter difficulties with capturing long-term correlations. Therefore, in the last two years, a new type of prediction model is outperforming the traditional prediction models in the field of time series prediction: the transformer. Its model structure is purely based on the attention mechanism, which determines the relevant information for each output while diminishing irrelevant information. The weight given to a specific input for a specific output is based on a similarity score between the input and output features. Consequently, the transformer is insusceptible to the limitations inherent in conventional time series models. However, it is yet unknown whether this model applies to long prediction horizons.

Therefore, this research proposes a long-term traffic flow prediction model that incorporates temporal and external features in a transformer structure. To synthesize this prediction model, the focus throughout this research lies on, (1) identifying important external features, and (2) investigating the effect of the transformer on longer horizons.

For commercial feasibility, a widely applicable prediction model is desired. Therefore, to test the genericity of the prediction models, all analyses are conducted for two locations, which are subject to different traffic behavior. The first is located on the ring road of Haarlem and is mainly affected by commuter traffic, whereas the second is located on the road to the

coast and has more irregular behavior. In addition, two state-of-the-art prediction models, a random forest and multi-layer perceptron are implemented, to provide a baseline prediction.

Multiple correlation analyses are performed to assess the importance of external features on the described locations. These demonstrate that the important features are location specific and include multiple time features, such as the hour of the day, day of the week, and the season. In addition, categorical features, such as whether a day is a national holiday or school vacation have a significant impact. At last, the temperature, radiation, and relative humidity are identified as important weather features.

Results show that the transformer outperforms the baseline prediction models on both short and long horizons, especially when the location is subject to irregular behavior. In addition, the positive impact of the external features on the prediction model competence is clearly shown, and the genericity of the model is highlighted by its applicability to multiple locations.

Table of Contents

Acknowledgements

This thesis concludes my time as a student at the TU Delft. During my research, I have had lots of support from various people, whom I would like to thank. First of all, I would like to thank my supervisor Alexander Koek for the weekly meetings in which he shared his experience, enthusiasm, and critical eye which has certainly been conducive for my research. I also really appreciate the involved questions regarding my weekend and the reassurance that it is normal to achieve less some weeks than hoped for. In addition, I would like to thank my supervisor Bart De Schutter for his constructive feedback and his high expectations both in terms of content and grammar. You have pushed me to always deliver qualitative documents and prepare efficient meetings. Moreover, I am still surprised that even though I performed many checks, you were always able to notice grammar mistakes at a glance.

Furthermore, I would like to thank my fellow students at 3ME, with whom I have spent many fun coffee breaks. Thereby also a big thanks to oras for all the free coffee, maybe someday the lost coffee card will magically reappear. In addition, a special thank you to Oyono for being so supportive and enthusiastic about my research. Writing these acknowledgements makes me realize that we are actually almost finished and I cannot wait to go on a road trip in the upcoming months.

Most of all I would like to thank Coco, Barbara, Benthe, Kirsten, and Myrthe, for being the best roommates. I know for sure that I could have not spent this much time with a lot of people as I did with you guys last year. I enjoyed the coffee laps we walked at eight o'clock sharp in the morning, to come back twenty minutes later at our house that turned into the office of the day. I enjoyed studying with the door open, so nothing would go unnoticed and every one of you would come by my room for a chat. I even enjoyed running at the end of the day. However, what I appreciated most is that after a long day there was always someone to talk through your day with, with a glass of wine on the sofa.

These are just a few things that have made the last year pass this fast and be so much fun. If you are reading this, you are probably one of the people that has made it such a nice year as well, so thank you!

Delft, University of Technology — C.A.M. Petsch
February 16, 2022

Chapter 1

Introduction

Traffic jams, overcrowded public transport, and limited parking availability are known daily problems in our transportation network, which cause long travel times and travelers discomfort. These issues will become even more serious as the traffic demand keeps increasing due to population growth and more intensive use of vehicles [13]. Also in the Netherlands, this is a well-known issue. Historical data on the traffic performance is provided by the Central Bureau for Statistics and reveals that from 2018 to 2019 the number of road vehicles has increased by 1.8%. In addition, the average distance traveled per vehicle has risen by 0.7% [6, 7]. Therefore, to cope with the increasing pressure on transportation networks, innovative solutions should be found to improve, extend, and make more efficient use of the network.

In the past decades, the main focus laid on expanding the transportation network. Unfortunately, this will not increase the capacity of the road network enough to meet the rising traffic demand [42]. An additional solution is to make more efficient use of the multi-modal transportation network by distributing travelers over different transportation modes, including new emerging modes based on shared vehicles. This will induce a shift in modality, and can be achieved in different ways [27]. Travelers should be informed beforehand about different possible itineraries, including options that combine different modes of transport. In addition, once insights are obtained about travel behavior, crowd flows can be influenced by authorities, for instance by dynamic pricing of public transport or parking lots.

1-1 Relevance of long-term traffic flow predictions

Recently, a demand for innovative solutions has come from multiple municipalities in the Netherlands. In 2020, the municipality of Amsterdam started Scale Up. This initiative aims to prevent large traffic flows to locations predisposed to attract busy crowds, also known as hotspots, by distributing travelers over time, locations, and modalities [19]. Therefore, insights into the location of the hotspots under different external influences, such as weather conditions, holidays, and events are required, such that traffic flows can be influenced.

In 2021, the municipality of Flevoland and the ministry of economic affairs and climate, started a project to increase the network accessibility during large events, such as the Floriade, Defcon, and Lowlands [48]. It is imperative to first gain insights into locations that restrict the network efficiency and traffic behavior under irregular circumstances. Next, measures can be taken accordingly, e.g., the deployment of extra transportation possibilities between locations, or giving priority to vehicles traveling in certain directions.

These initiatives highlight the necessity of an adequate traffic flow prediction model, which takes external influences into account. This way, valuable insights into traffic behavior can be implemented to make more efficient use of the transportation network, increase accessibility, and provide multi-modal itineraries. This may ultimately decrease the pressure on the transportation network and increase traveler comfort.

This research focuses on a fraction of the entire transportation network by making a long-term traffic flow prediction of vehicles on the main roads. The global objective is to combine multiple transportation modes into one network. Therefore, the focus lies on areas where different transportation modes intersect and the main roads connecting them. Current research on traffic flow prediction often considers control of intelligent traffic systems, for which short-term predictions based on real-time measurements are required [35]. However, to be able to inform travelers and authorities, a longer prediction horizon is required. The exact definition of short- and long-term differs throughout state-of-the-art literature, in which the division lies somewhere between a prediction horizon of 30 minutes and several hours [32, 33, 55]. In this research, a prediction is defined as long-term when the prediction horizon exceeds one hour. In addition, the maximum required horizon is assumed to be 24 hours, divided into slots of one hour, such that the prediction of the next day is composed of 24 predictions.

1-2 Challenges of long-term traffic flow predictions

The complex, nonlinear nature of traffic flows makes it difficult to make accurate long-term predictions [36]. Therefore, the main challenge that arises with the increased prediction horizon is the increase in uncertainty. By implementing multistep autoregressive predictions, errors propagate through the network and accumulate with time [60]. On the other hand, it is difficult to make accurate long-term single-step predictions, due to the large gap in time between the prediction and the most recent available data [35, 69, 72]. To limit the amount of uncertainty, it is important to capture as many features describing traffic flow behavior as possible. Therefore, it is important to investigate which features influence traffic behavior.

Recent research has shown that to describe traffic flow behavior, it is important to capture both temporal and spatial features [23, 36, 74, 76]. Temporal features describe the correlations of traffic flow throughout time and spatial features describe the correlations throughout the road network. To clarify, the rush hour starting at approximately the same time every working day at a certain location can be related to a temporal feature. On the other hand, congestion at a certain location in the network will influence other parts of the network, which can be seen as a spatial feature. For the scope of this research, the spatial features are excluded but they could be included in future studies, to extend the prediction model.

In addition to temporal and spatial features, traffic behavior is influenced by external features such as events, the weather, and construction works, as shown in [14, 15, 38, 43, 75, 77]. As an illustration, the traffic flow to the coast will be significantly different on a sunny day in contrast to a rainy day. Moreover, the yearly festival Lowlands attracts approximately 55,000 visitors, influencing the traffic flow accordingly. Which external features are important differs per location, prediction horizon, the hour of the day, etc. However, for implementation purposes, it is desired to obtain a generic prediction model framework, which can be utilized for different locations.

Therefore, it is important to identify relevant external features, such that these can be incorporated in the prediction model while maintaining genericity by taking the applicability to multiple locations into account.

1-3 Incentive behind traffic flow prediction models

Various prediction models have already been applied to traffic prediction tasks. Recently, a lot of research is focused on implementing machine learning methods to traffic flow predictions, which are shown to outperform more classic prediction methods [18, 36, 76]. These machine learning methods have been divided into two categories, based on the input data being processed. Classic machine learning models implement input features corresponding to a single timestamp. Nevertheless, traffic flow is known to have a sequential behavior, which will be referred to as auto-correlation. More advanced methods allow us to consider these characteristics by looking into input features of different timestamps. These methods are shown to be beneficial in terms of performance for short-term traffic predictions [22, 36, 66]. However, whether the inclusion of auto-correlation is advantageous on longer prediction horizons is yet unknown.

Models designed to capture auto-correlation in sequential data are traditionally based on recurrent layers. These include broadly-known models such as the long short-term memory model and the gated recurrent unit [11, 36, 51, 74, 76]. However, these methods have some inherent limitations due to the recurrent structure. First, parallel computations can not be done, which makes them computationally expensive. Moreover, the models encounter difficulties with capturing long-term correlations. At last, the model dynamics are time-invariant, whereas traffic flow is not.

To address these limitations, in the last two years, a novel prediction model known as the transformer, first proposed in [58] for natural language processing, is increasingly implemented in time series prediction tasks and is shown to outperform the recurrence-based models [21, 34, 67]. The transformer is insusceptible to the limitations inherent in conventional time series models, because the model structure is purely based on the attention mechanism, first proposed in [1], which suggests a structure that searches for relevant input information, while diminishing irrelevant information. The weight given to a specific input for a specific output is based on a similarity score between the input and output features. First, during training, all computations can be made in parallel, making it computationally more efficient. As an illustration, in [67] and [70] the training time is shown to decrease approximately 14 times when implementing a transformer instead of a gated recurrent unit. There should not be emphasized too much on this exact number, since this will differ for each data set

and application. Secondly, the maximum path length between an output and an input is decreased. According to [28], the larger the path between two variables, the harder it is for the model to learn correlations. Therefore, the transformer encounters fewer difficulties regarding long-term correlations [15]. Finally, because the transformer model parameters are based on the input feature, the model dynamics vary over time. This highlights why models based on the attention mechanism are gaining popularity in the field of traffic predictions.

The disadvantage of machine learning methods is that the model obtained is a black-box model, which means that the model behavior is not intuitive. In addition, whether the transformer model is suitable for long horizons still has to be investigated. Therefore, it is important to compare the predictions with the predictions of one, or multiple baseline prediction models. In state-of-the-art literature, the random forest and multilayer perceptron (MLP) are often implemented for this purpose [11, 31, 36, 39]. Because these models do not take previous traffic flow into account, they belong to the first category of machine learning models described above.

1-4 Research objectives

As highlighted in the previous sections, there is a necessity for a long-term traffic flow prediction model, which considers temporal, spatial, and external features. Therefore, the goal of this thesis is formulated as follows:

Develop a generic long-term traffic flow prediction model that can predict 24 hours ahead and that incorporates temporal and external features in a transformer-based framework.

To synthesize the proposed traffic flow prediction model, the following research questions will be answered, which align with the previously induced challenges:

1. Which external features are important to capture in long-term traffic flow prediction? In addition, how can these be incorporated into a generic prediction model?
2. What is the effect of implementing the transformer on different prediction horizons for traffic flow predictions?

1-5 Outline of this thesis

To answer the research questions described above, the rest of this thesis is structured as follows. Chapter 2 discusses the preliminary background of methods used in this research. First, different correlation analysis techniques are discussed, which are implemented to investigate relevant input features. Next, the working principles of the baseline prediction models and the transformer are highlighted. Chapter 3 describes the case study on which the correlation analysis and prediction models are applied. Moreover, the available data are discussed and data processing and preparation steps are taken. Chapter 4 investigates the correlations in the data set, to gain insights into the importance of different features on traffic flow. When the final input feature set is obtained, the baseline models and transformer are implemented in Chapter 5 and Chapter 6, respectively. The performance of these prediction models is discussed and compared in Chapter 7. Finally, in Chapter 8 conclusions are presented, and future recommendations are discussed.

Chapter 2

Theory behind traffic flow predictions

This chapter elaborates on the decisions made regarding methods used throughout the rest of this research and the corresponding theoretical background. First, multiple correlation analysis techniques are described that can be used to identify important features. Next, the general implementation of the baseline prediction models and transformers is discussed. Next, there is elaborated on the working principles and model structures.

2-1 Correlation analysis techniques

As discussed in the introduction, recent research has shown that to describe traffic flow behavior, it is important to capture temporal features [23, 36, 74, 76]. In addition, traffic behavior is correlated with external features such as events, the weather, and construction works, as shown in [14, 15, 38, 43, 75, 77]. Therefore, it is important to incorporate these features in the prediction model.

In general, the size and computational effort of prediction models grow with an increase in the number of input features. Therefore, it is undesired to input insignificant features. This highlights the necessity to make a hypothesis on which features are important for the prediction model, which can be done with correlation analyses.

Two different correlation analyses are performed on the data set. First, by clustering days, the similarity between days is investigated. Secondly, cross-correlation techniques can be used to investigate the correlation between two different data sets. This is implemented to determine the cross-correlation between traffic flow and external features. In addition, it is used to look into the auto-correlation in traffic flow.

In state-of-the-art literature, different correlations are found for different locations, which indicates that the correlation analyses are location specific and should be reconsidered when the prediction model is applied to a different location [63, 68].

Table 2-1: K-means and agglomerative hierarchical clustering characteristics, where + and − indicate a positive and a negative property, respectively.

	K-means clustering		Agglomerative hierarchical clustering
+	Scales well to large data sets	+	Finds a global optimum
−	Requires the number of clusters to be specified	+	Number of clusters can be chosen based on the dendrogram
−	Sensitive to local minima	−	Computationally more expensive
−	Assumes data is separated in sphere-like clusters		

2-1-1 Clustering techniques

To look into similarities between different days, the data set with data at each time interval is transformed into features corresponding to the entire traffic flow of one day. As a result, each day is represented by one vector, and clustering techniques can be applied. There are two main methods often implemented in state-of-the-art literature to find similarities in traffic flow behavior: k-means clustering [68] and agglomerative hierarchical clustering [49, 63, 68].

Comparison of k-means- and agglomerative hierarchical clustering

In Table 2-1, the advantages and disadvantages of both clustering methods are shown. The main downside of k-means clustering is that it requires specifying the number of clusters beforehand. However, because clustering is implemented to analyze the data, this number is unknown. In addition, it is required to run the algorithm multiple times, because it is sensitive to local minima. At last, the algorithm experiences difficulties with clusters of different sizes, or non-spherical shapes. On the contrary, agglomerative hierarchical clustering is computationally more expensive, but it is not subject to the limitations inherent in k-means clustering. Therefore, it is chosen to implement agglomerative hierarchical clustering.

Agglomerative hierarchical clustering

Agglomerative hierarchical clustering is a bottom-up approach. It starts by assigning each data point to a separate cluster. Next, the closest two clusters are identified and merged. This continues until all data points belong to the same cluster. The algorithm contains two, to be specified metrics; the distance metric and the linkage criteria. The distance metric is used to calculate the similarity between two clusters and is often chosen as the Euclidean distance. The linkage criterion defines how the distance is computed. For instance, single-linkage calculates the distance between the most similar parts of two clusters and average-linkage calculates the average distance of each point in one cluster to every point in the other cluster. A regularly implemented linkage criterion is Ward's linkage [61], which minimizes the total within-cluster variance. Ward's method calculates the increase in cluster variance by merging cluster A and B as

$$I_{AB} = \frac{n_a n_b}{n_a + n_b} \left(\bar{y}_a - \bar{y}_b\right)^2, \tag{2-1}$$

where n_a and n_b represent the number of data points in cluster A and B, respectively. In addition, $\bar{y}_a$ and $\bar{y}_b$ are the cluster centers. The increase in cluster variance is calculated for all clusters. Subsequently, the cluster with the smallest increase in squared distance is merged and the cycle continues. For further clarifications, the mathematical derivation can be found in [61]. The advantage of Ward's method can be clearly shown by rewriting (2-1) as

$$I_{AB} = \frac{1}{\frac{1}{n_a} + \frac{1}{n_b}} \left(\bar{y}_a - \bar{y}_b\right)^2, \tag{2-2}$$

which shows that when n_a and n_b increase, I_{AB} also increases. Therefore, Ward's method is more likely to merge clusters of smaller or equal size, and it will tend to avoid many small-sized clusters. In the end, the algorithm outputs a dendrogram, which indicates the hierarchical relation between the clusters and illustrates the effect of an increase and decrease in the cluster granularity level. By slicing the dendrogram horizontally, the clusters are formed based on the corresponding increase in cluster dissimilarity and number of clusters.

2-1-2 Cross-correlation techniques

To investigate the relation between two different data sets, cross-correlation techniques can be applied. Two methods often implemented are: Pearson's method [68] and Spearman's rank method [31], which find a linear and a nonlinear correlation, respectively.

Pearson's correlation method

The correlation between data set x and y, of length n, can be found by fitting a linear line through the data. The strength of the correlation, is represented by Pearson's correlation coefficient (r_{pearson}), which indicates the deviation between the data and the fitted line. The correlation coefficient lies between -1 and 1, where -1 denotes a strong negative correlation and 1 a strong positive correlation. Pearson's correlation coefficient is calculated by dividing the covariance of x and y by the standard deviations, s_x and s_y as

$$\begin{aligned} r_{\text{pearson}}(x,y) &= \frac{\operatorname{cov}(x,y)}{s_x s_y} \\ &= \frac{n\sum_{i=1}^{n}(x_i y_i) - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{\left(n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right)\left(n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2\right)}} \end{aligned} \tag{2-3}$$

The derivation of the formula above can be found in [45]. It should be noted that the coefficient is not correlated to the slope of the fitted line.

Spearman's rank correlation method

Spearman's rank builds further upon Pearson's method. It is able to find a nonlinear correlation, under the assumption that the data is monotonic. In addition, Spearman's rank

coefficient also varies between -1 and 1. First, the rank of each data point $x_{\mathrm{r},i}$ corresponding to x_i is determined, which varies between 0 and 1. Then, instead of the actual values, Spearman's rank method investigates the correlation on the ranked data set, and the correlation coefficient is calculated as

$$r_{\text{spearman}}(x, y) = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n\left(n^2 - 1\right)}, \tag{2-4}$$

where $d_i^2 = (x_{\mathrm{r},i} - y_{\mathrm{r},i})^2$ equals the squared distance between the rank of two data points. For the derivation from Pearson's correlation coefficient to Spearman's rank coefficient, there is referred to [53].

Adequacy of cross-correlation techniques in traffic flow analyses

The cross-correlation techniques described above are implemented in traffic flow analyses to find the relation between traffic flow and another time series data set, such as the temperature. In addition, these methods can be used to investigate the auto-correlation in traffic flow. By shifting the traffic data multiple times, the correlation between previous and future time instances can be investigated, providing information about whether certain previous traffic flow measurements might be important features for the to be predicted traffic flow [31].

2-2 Implementation of the prediction models

In Chapter 1, the incentive behind the transformer and the baseline prediction models are discussed. It should be noted that other regression methods could also be implemented as a baseline. However, for the scope of this research, the number of prediction models is limited to these. This section focuses on the general implementation of the prediction models. Subsequently, the specifics of the baseline prediction models and transformer are discussed in the next sections.

The core idea of the prediction models is to map an input sequence x to a predicted output sequence $\tilde{y}$, which matches the actual output y. Unfortunately, one of the main challenges is that the model can correspond too closely to the historical data, which is referred to as overfitting. Therefore, first, it is important to divide the available data into a train and test set. This is done, to be able to test the prediction model on a data set that is not seen by the prediction model before. In addition, a part of the training set is set apart as the validation set, to investigate, during training and tuning, whether the model might be overfitting. To elaborate, training the prediction models is done on the training set. After each step, the model is evaluated on the validation set, to indicate whether the model has a similar behavior on unseen data. Next, the prediction model can be adjusted based on these preliminary results. The difference between the validation and test set is that the final prediction model is influenced by the first, due to the adjustments.

How the final prediction is made, depends on the model parameters of the prediction models. These are composed of two types of model parameters; trainable parameters and hyperparameters. The first are optimized during training. In addition, hyperparameters are parameters

that have an influence on the final model performance but are not optimized during the training process. Therefore, these have to be investigated and optimized separately. How the final prediction models are obtained, is described by utilizing these two types of model parameters and is briefly discussed in the next sections.

2-2-1 Training the prediction models

The conversion of the input to the output is done by a sequence of computations. The specific computations depend on the structure of the prediction models and will be discussed in the next sections. However, these are all based on a set of trainable variables. These variables are optimized during training, by minimizing the to-be-specified loss function L, which compares the predicted output $\tilde{y}$ with the actual output y. In the multilayer perceptron (MLP) and transformer, this is done by backpropagation, first proposed in [50], in which the derivatives of the loss function with respect to the weights are calculated, for further details there is referred to [20]. Moreover, the optimization algorithm used is Adam, first proposed in [30], because it has become the state-of-the-art in current literature [15, 34, 67]. On the other hand, the classification and regression tree algorithm is used to train the random forest, by minimizing the loss function in the nodes, weighted by the number of samples.

The most common loss functions for regression applications are investigated. If the output is a Gaussian distribution, the mean squared error is the preferred loss function. However, if the output is subject to more outliers, the mean absolute error (MAE) can be implemented which calculates the absolute error between the actual and predicted output [20]. In state-of-the-art literature regarding traffic flow predictions, both loss functions are commonly implemented [5, 16, 22, 24]. Because it is desired to punish large errors, the mean squared error is assumed to be applicable in this research.

2-2-2 Hyperparameter optimization

The idea of hyperparameter optimization is to define a space for each hyperparameter, investigate the model performance corresponding to different hyperparameter combinations, and choose the hyperparameters with the best model performance on a validation set. Three methods often used are, grid search, randomized search, and Bayesian hyperparameter optimization [3, 20]. The first investigates all possible combinations, which guarantees to find the optimal combination but is computationally expensive. The second investigates a random number of combinations, which allows extending the search space, without increasing the number of computations. As opposed to the other two methods, Bayesian hyperparameter optimization does not randomly choose evaluated hyperparameters, but bases this on knowledge of previous evaluations. Therefore, it is likely to reach the optimal parameters faster and is often shown to be the preferred method [2, 29, 52]. The idea is to make fewer calls to the objective function, on the cost of spending a bit more time on selecting the next parameters. In [4], the extra time spent on selecting the parameters is shown to be negligible. As an illustration, finding the next set of parameters only took a few seconds, whereas evaluating the objective function took hours. Therefore, Bayesian hyperparameter optimization will be implemented.

Two important criteria in Bayesian hyperparameter optimization are the surrogate model, which is a representation of the objective function, and the selection function. It is chosen to use Hyperopt [4], because it is known to be user-friendly. The tree parzen estimator is implemented as the surrogate model, because it has state-of-the-art performance and can incorporate all different kinds of variables [20, 71]. In addition, the maximum expected improvement is chosen as the selection function, because it has a good performance, is easy to evaluate, and is inexpensive to compute [17, 71]. The mathematical derivations behind the optimization algorithm can be found in [2].

2-3 Baseline prediction models for traffic flow predictions

The main principles of the baseline prediction models are briefly elaborated on. It is chosen not to go into much detail because these models are commonly used machine learning models, which have been explained in great depth quite often. Therefore, only the specifics for this research are highlighted and if desired; more thorough explanations can be found in [20].

2-3-1 Random forest

The random forest is an ensemble method based on multiple decision trees. The main idea is to iteratively split the data set into two subsets, based on a single feature and threshold, with the objective to minimize the mean squared error. The prediction made by a node then corresponds to the average of the samples in the specific node and is referred to as the value.

A disadvantage of the decision tree is that it is sensitive to small variations in the data. Therefore, the random forest is more commonly implemented, which is an ensemble method of multiple decision trees, in which the multiple predictions are averaged. Different trees are designed by looking into a random subset, instead of the entire set of features and thresholds, to base the split on. A positive incidental is that the relative feature importance can be easily calculated by the weighted decrease in each node mean squared error, through which information about the model behavior can be extracted.

Important hyperparameters for the random forest are the maximum tree depth, the minimum samples per leaf, and the number of trees. The first constrains the random forest because a deeper decision tree can fit more complicated functions. In addition, the second constrains the random forest by prohibiting nodes with only a limited number of samples. The last hyperparameter determines the number of decision trees in the random forest because by increasing the number of trees the chance of overfitting will decrease.

2-3-2 Multilayer perceptron

The MLP is a classic neural network, which is based on one input layer, multiple hidden layers, and an output layer. Each hidden layer is composed of a to be specified number of neurons and is subject to a nonlinear activation function, which is chosen to be the Rectified Linear Unit (ReLU), because it is shown to be computationally fast and to work well [20, 71]. Moreover, the input is concatenated with the output of the last hidden layer, as proposed

in [8], and they are inputted to the output layer. As a result, simple patterns will not get distorted through the sequence of computations.

The structure of the MLP depends on the number of hidden layers and the number of neurons in each layer. Furthermore, the training algorithm depends on two additional hyperparameters; the learning rate and batch size. These determine the step size and number of data samples processed at each iteration, respectively.

2-4 Transformers for traffic flow predictions

The transformer, initially proposed for natural language processing [58], is shown to be applicable in many other topics and is implemented for time series predictions in among others [15, 34, 67]. To be able to understand the type of transformer required in time series prediction, there will be elaborated on the transformer structure and the implementation.

2-4-1 Encoder-decoder structure

The transformer is based on the encoder-decoder structure, first proposed in [9]. This structure is implemented to map an input sequence to an output sequence of a different length. In some studies, this is referred to as the sequence to sequence structure. However, to avoid misconceptions, only encoder-decoder will be used throughout this research. This structure is composed of two components, as shown in Figure 2-1. At time step t, the encoder converts the encoder inputs $x_{\text{enctot},t} = [x_{\text{enc},t-l}^T \dots x_{\text{enc},t}^T]^T$, where l denotes the number of previous steps taken into account by the model. In addition, each input is composed of the traffic flow y and additional input features x, corresponding to the specific time stamp as

$$x_{\text{enc},t-l} = \begin{bmatrix} y_{t-l} & x_{t-l}^T \end{bmatrix}^T \tag{2-5}$$

Next, the decoder uses the encoder output and decoder inputs, to predict $\tilde{y}_{\text{tot},t} = [\tilde{y}_{t+1} \dots \tilde{y}_{t+h}]^T$, where h equals the maximum prediction horizon and $\tilde{y}_{t+i}$ the traffic flow prediction at $t+i$ for $i = 1, \dots, h$. The advantage of this prediction structure is that the additional input features, corresponding to future timestamps, can be taken into account through the decoder inputs, which are set up as

$$x_{\text{dec},t+h} = \begin{bmatrix} \tilde{y}_{t+h-1} & x_{t+h}^T \end{bmatrix}^T, \tag{2-6}$$

where the previously predicted traffic flow is inserted, because future traffic flow is unknown. How these inputs are converted, depends on the model structures used inside the encoder and decoder. The encoder and decoder were historically often based on recurrence, whereas the transformer is based on the attention mechanism.

2-4-2 Transformer structure

The structure of the transformer, as proposed in [58], is shown in Figure 2-2c. The model is composed of (masked) multi-head attention blocks, positional encoding, add and normalize layers, and feedforward layers, which will all be shortly elaborated on below. Each encoder and decoder layer can be stacked N times to extract even more information.

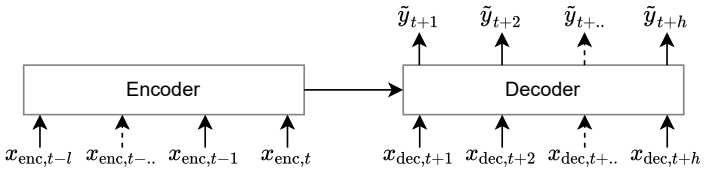


Figure 2-1: Schematic overview of the encoder-decoder structure

(Masked) multi-head attention

The main component of the transformer is the multi-head attention block, which is used in three different places. First, self-attention is implemented in the encoder. In addition, masked self-attention is implemented in the decoder. At last, attention is implemented between the encoder and decoder.

The principle of attention is to represent an input, by a weighted sum of a sequence, to include relevant information of the entire sequence in the specific input. How much attention is given to input j to represent input i, is specified by the attention weight $\alpha_{i,j}$, which is calculated by taking the softmax of an attention score $e_{i,j}$, to ensure that the score is scaled.

There are two kinds of attention mechanisms most commonly used in literature, each based on another scoring function. The first is additive scoring, which is sometimes referred to as Bahdanau scoring and was first proposed in [1]. The second is implemented in the transformer and is based on the scaled dot scoring function, as proposed in [58]. The disadvantage of the first is that it requires many computations to compute the attention weights, and is consequently less frequently implemented. Therefore, there is only elaborated on the scaled dot scoring function.

In self-attention, the attention score is based on one specific input vector, whereas for the attention between the encoder and decoder, the attention score is based on two different vectors. However, the working principles are equivalent. The input vector is used in three ways. By comparing $x_{1,i}$ to another input $x_{2,j}$, a weight is established for output y_i and y_j. In addition, the output vector is calculated as a weighted sum of the input vector. These are called the query (q), key (k), and value (v), respectively. The input vector is linearly transformed by the trainable matrices W_q, W_k, and W_v, into dimension d, to obtain these vectors as

$$q_i = W_q x_{1,i}, \quad k_i = W_k x_{2,i}, \quad v_i = W_v x_{2,i}, \tag{2-7}$$

where for self-attention $x_1 = x_2$. Subsequently, the similarity measure is taken, to calculate the attention score between the query and the key, such that

$$\alpha_{i,j} = \text{softmax}\left(\frac{q_i k_j^T}{\sqrt{d}}\right), \tag{2-8}$$

and the output is calculated as

$$\tilde{y}_i = \sum_{j=1}^{l_x} \alpha_{i,j} v_j, \tag{2-9}$$

where l_x is equal to the length of the value vector. The scaling factor $\frac{1}{\sqrt{d}}$ is applied to ensure that the dot product does not go into regions where the softmax has small gradients [58]. The advantage of this approach is that matrix multiplications can be implemented. As a result, the output can be calculated as

$$\tilde{y} = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{2-10}$$

The calculation of the output $\tilde{y}$ is schematically shown in Figure 2-2a. In the decoder, leftward information should be prevented, because future traffic flow inputs are unknown. All future inputs are therefore set to $-\infty$ inside the softmax function. Consequently, these inputs will vanish, which is referred to as masked attention.

Instead of transforming the input sequence once, in [58] it is found that it is more efficient to project the query, keys, and values multiple times into a different dimension. Afterwards, each projection goes through an attention mechanism in parallel. Next, the independent outputs are concatenated and again linearly transformed into a final output, as shown in Figure 2-2b. The final output of the multi-head attention is calculated as

$$\tilde{y}_{\text{concat}} = W_0\left(||_{k=1}^{K}\tilde{y}_{\text{head},k}\right) + b_0, \tag{2-11}$$

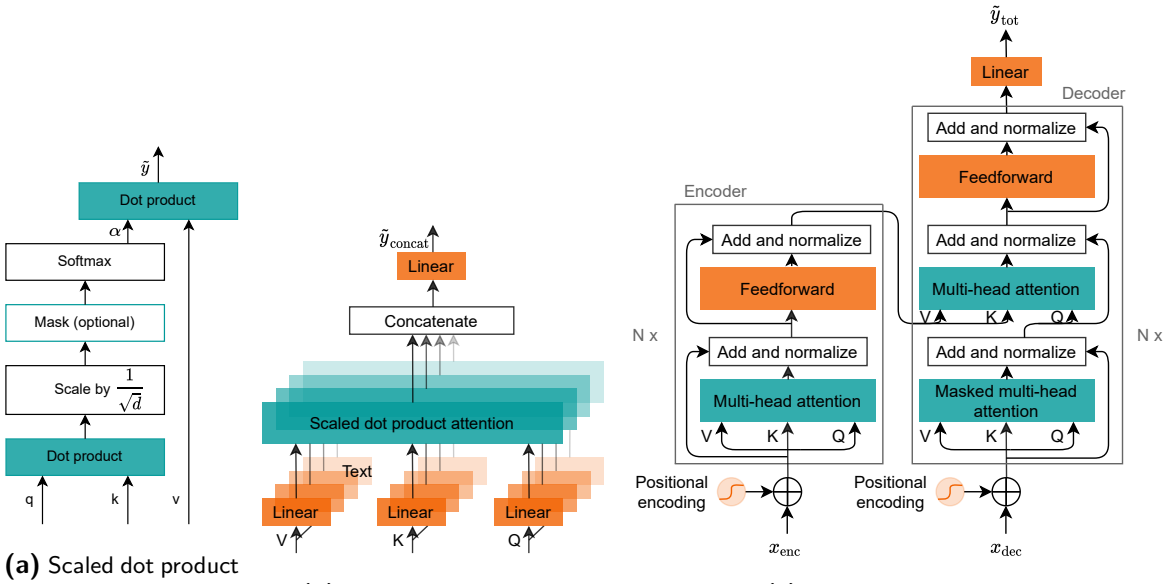
in which $||_{k=1}^{K}$ represents the concatenation of the K output sequences of $\tilde{y}_{\text{head},k}$. In addition, the concatenated vector is linearly projected by trainable matrices W_0 and b_0. The output of each head is based on a different query, key, and value. To ensure that the computational cost of the multiple heads is similar to that of a single head, the dimension of the query, key, and value is decreased to $\frac{d}{K}$. Consequently, the total number of trainable parameters is similar to that of a single attention head [58]. The idea behind this approach is that multiple attention mechanisms allow for different projections and other dependencies to be captured, such as shorter- or longer-term dependencies, without increasing the complexity [15].

Positional encoding

By omitting the recurrence in the network, all information regarding the order of the input sequence is lost. This is undesired because the sequence order still contains important information. Therefore, positional encoding is implemented, which includes information regarding the relative position of the data in the input. In the original paper [58], sine and cosine functions of different frequencies are added to each input dimension, such that close data points differ in higher frequencies [24, 37]. However, the positional encoding requirements for time series predictions differ from natural language processing.

The objective is to provide information about the relative position. The simplest approach is to concatenate a vector ranging from 0-1 to the input. In this research, the number of input features taken into account is fixed. Therefore, concatenating this vector, referred to as the age feature, is sufficient to represent the relative order of the input [5, 34]. On the other hand, this method is unfeasible in natural language processing because the input dimension varies.

On the top of the relative position, additional global positions are important in time series predictions, such as the month of the year or the day of the week (dow). Therefore, some



(a) Scaled dot product attention

(b) Multi-head attention

(c) Transformer structure

Figure 2-2: Schematic representation of the scaled dot product attention, multi-head attention, and the transformer in (a), (b), and (c), respectively.

researchers concatenate additional trainable layers, which learn periodicity in the data. Subsequently, these are inputted to the prediction model [21, 34, 67]. However, this requires additional trainable model parameters and can give rise to inserting non-intuitive periodicity. Moreover, important periodicity can also be identified in correlation analyses. Therefore, it is chosen to only concatenate the age vector during positional encoding and investigate the additional global positions in the correlation analyses.

Add and normalize layers

Residual connections are implemented by adding the output of the attention layer or feed-forward layer and the original input, and the result is normalized. As a result, gradients are allowed to flow directly through the network, which improves the training ability [25].

Feedforward layers

The feedforward layer linearly projects its input two times, with a ReLU activation function in between, which outputs the input if positive and zero otherwise. The parameters for this projection are optimized during training, to process the output of an attention layer such that it is more suitable as an input for the next attention layer. At last, a single linear layer is implemented which transforms the decoder output to the desired output dimension.

2-4-3 Implementation of the transformer

Recall that the idea of a prediction model is to map an input sequence x to a corresponding predicted output sequence $\tilde{y}$, through a sequence of computations. The specific computations

for the transformer have been explained in the section above and are based on a set of trainable variables. The total number of trainable variables depend on a few parameters, the dimension of the feedforward layer d_{ff}, the number of encoder and decoder layers N, the dimension of the input d_x, and the dimension of the output d_y, such that the total number of trainable variables equals

$$n_{\text{transformer}} = n_{\text{enc}} + n_{\text{dec}} + n_{\text{output}}, \tag{2-12}$$

with

$$\begin{aligned} n_{\text{enc}} &= N \cdot (1\,(4\,(d_x \cdot d_x + d_x)) + 2\,(d_x + d_x) + 1\,(d_x \cdot d_{\text{ff}} + d_{\text{ff}} + d_{\text{ff}} \cdot d_x + d_x)) \\ n_{\text{dec}} &= N \cdot (2\,(4\,(d_x \cdot d_x + d_x)) + 3\,(d_x + d_x) + 1\,(d_x \cdot d_{\text{ff}} + d_{\text{ff}} + d_{\text{ff}} \cdot d_x + d_x)) \\ n_{\text{output}} &= d_x \cdot d_y + d_y \end{aligned} \tag{2-13}$$

These illustrate that the number of trainable parameters is highly dependent on the dimension of the input. For the derivations of these equations, there is referred to Appendix A-1.

As mentioned in Section 1-3 one of the advantages of the transformer is that training can be done in parallel, which significantly decreases the computational complexity. In the encoder, all required inputs are known from the start. However, in the decoder, the previously predicted output is used to compute the consecutive output. To allow for parallel computations, teacher forcing is applied in the decoder [20]. Instead of providing the predicted output to the next decoder input, the model provides the known output from the training set. Therefore, the gradient of the loss function can be computed separately for each layer.

The calculations above already highlight a few hyperparameters, which influence the number of trainable parameters; the number of layers (N) and the dimension of the feedforward layer (d_{ff}). In addition, the number of attention heads (n_{heads}), the learning rate l_r, and batch size are also hyperparameters in the transformer.

2-5 Summary

This chapter has elaborated on the decisions for specific methods used throughout the rest of this research and the corresponding theoretical background. First, state-of-the-art literature has shown that to describe traffic flow behavior, it is important to capture temporal, spatial, and external features. In general, the computational effort of prediction models grows with an increase in the number of input features. This highlights the necessity of identifying important features, which can be done by implementing correlation analyses.

Clustering methods can be implemented to group similar days. Different methods are compared, and agglomerative hierarchical clustering is found to be most suited because it does not require to specify the number of clusters beforehand. In addition, cross-correlation methods can be implemented to find correlations between two data sets. There is elaborated on Pearson's and Spearman's rank method, which find a linear and nonlinear relation, respectively. The adequacy of cross-correlation techniques in traffic flow analysis is that it allows finding correlations between traffic flow and external features, such as the temperature. In addition, the auto-correlation in traffic flow can be investigated.

Next, the incentive behind the transformer in traffic flow predictions stems from three main advantages. First, the structure allows for parallel computations, which makes it computationally efficient. Next, it does not encounter difficulties regarding long-term correlations,

because the maximum path between an output and input equals one. At last, the model is time-variant, because the parameters are based on the input feature. In addition, the choice for the random forest and MLP as the baseline models is highlighted.

The implementation of the prediction models is explained. Here, the choice for the root mean squared error (RMSE) as the loss function, Adam's method as the optimization algorithm, and Bayesian parameter optimization for the hyperparameter optimization are substantiated.

At last, the entire transformer structure and the implementation are described. Originally, the transformer was proposed for natural language processing. Therefore, the requirements opposed by implementing it on a time series predictions task are discussed. In the end, the scaled dot scoring function is chosen as the scoring function, and the age vector is thought to be sufficient for positional encoding in this research.

Chapter 3

Analysis of historical traffic flow and weather data

The data set used throughout this research is described and evaluated in this chapter. First, the available traffic data in the Netherlands is described in Section 3-1. Next, there is briefly elaborated on the technical background of the electromagnetic loop detector, the case study, and the available traffic and external data. Next, Section 3-2 describes the data preprocessing steps, divided into the set-up of the data set and data cleaning. The data is investigated, and quality checks are performed to clean the data and limit the effect of the following irregularities: the misalignment due to summer and winter time, erroneous and inaccurate traffic flow data, and missing data. Finally, the data is prepared in Section 3-3, by aggregating the data in hourly intervals, adding traffic flow of multiple lanes, and implementing feature scaling. Next, the final data set for all locations is compared, to group locations with similar behavior.

3-1 Available data

Traffic data can be divided into two categories, data generated by a system bound to the infrastructure and data generated in mobile systems, such as floating car data [26]. For the scope of this research, the prediction models discussed do not use the second type of data, and therefore, there is not further elaborated on these specifics. The push-button, vehicle selective detectors, and electromagnetic loop detectors are the most frequently used detectors of the first category in the Netherlands [64]. The first is used for pedestrians and traffic at bicycle lanes, the second recognizes priority vehicles such as ambulances or line buses, and the last detects all vehicles crossing the detector. Because the focus lies on the traffic of motor vehicles on the main roads, the last type of detector is important for this research.

This data is provided by two main sources in the Netherlands. First, the Nationaal Dataportaal Wegverkeer (NDW), a collaboration of the government, traffic authorities, and provinces, has made traffic data of over 17500 locations in the Netherlands publicly available since 2009,

and still updates this data every minute. Secondly, intelligent traffic control installations can provide traffic data. The disadvantage of the second is that the data is not automatically stored, but this has to be requested. Therefore, historical data of most locations is not available yet. Because the prediction models investigated in this research are based on historical data, the second data source has not been used in this research. However, if more information is required, it can be beneficial to fuse multiple data sources in future research.

3-1-1 Electromagnetic loop detector

The data provided by the NDW is gathered by electromagnetic loop detectors. These are composed of a copper coil connected to an electronic circuit and are installed in the road surface. Due to the alternating current in the coil, a magnetic field is generated. Whenever a metal object is present above the loop, the magnetic field is disturbed. The change in the magnetic field is noticed by the electronic circuit, compared to a certain threshold, and converted to a binary signal. Whenever the change exceeds this threshold, the system outputs 1, indicating that a vehicle is detected. The threshold is an important parameter of the electromagnetic loop detector because it determines the sensitivity of the detector. If the detector is too sensitive, a distortion in the magnetic field caused by passing vehicles on a neighboring lane can lead to false detection. For a more extensive explanation of the working principles, there is referred to [64].

A report written by Polman has investigated the properties of this detector, and states that the advantages are that they are often implemented, reliable, and not influenced by severe weather conditions [47]. In addition, an indication of the reliability of the detector is provided, which states that the accuracy equals approximately $95 - 98\%$ and the system is available $97 - 100\%$ at the time. Therefore, it is assumed that the data provided by the electromagnetic loop detectors is a good representation of the actual traffic flow. However, data analyses have to be performed to discover when the system is unavailable or inaccurate.

3-1-2 Case study and available traffic data

The global objective is to combine multiple transportation modes into one network and to investigate the effect of external influences on the traffic flow. Therefore, the focus lies on areas where different transportation modes intersect, and a location is chosen which is known to be highly influenced by these factors. Figure 3-1 shows the area of Haarlem, Bloemendaal, and Zandvoort. The blue circles indicate the locations of the available detectors in this area. Each circle represents detectors in both directions of the road, indicated by the number with or without an r. Monthly traffic data, aggregated at a five-minute interval, is provided for 2017, 2018, and 2019, at all the indicated locations. It is chosen not to include 2020, because the traffic behavior is significantly different from the other years due to the COVID-19 pandemic. Therefore, it is assumed that this data is not representative of the impending years. The monthly data are combined into one data set and separated based on the location.

Not all the provided information is useful for this research. To ensure that futile data does not have to be processed by the model, the data set is converted. Each lane at every location consists of six different measurements, providing data for vehicles in different categories. The vehicles are categorized based on the time a detector is occupied, which can be related to the

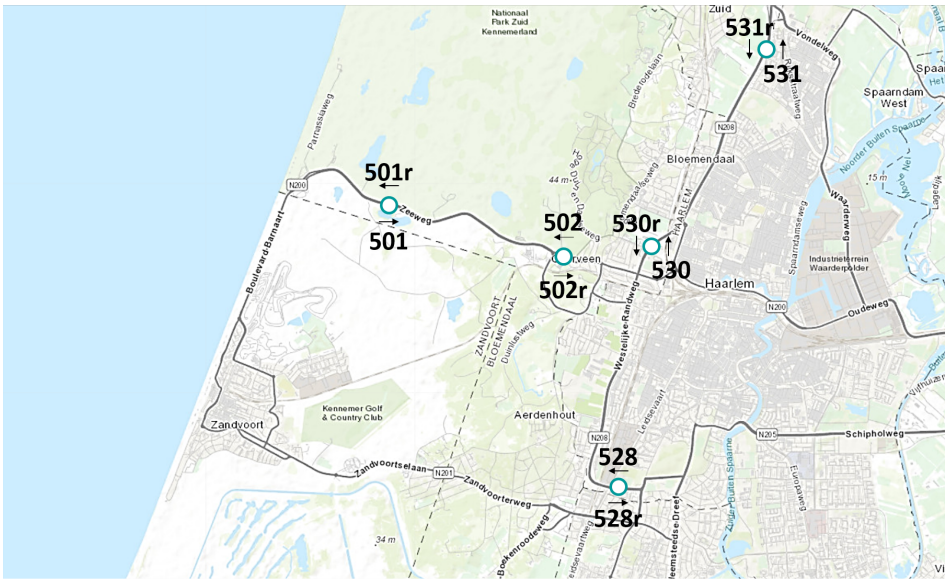


Figure 3-1: Roadmap of the area of Haarlem, Bloemendaal, and Zandvoort, in which the blue circles indicate the locations of the sensors.

length of the vehicle. Because the type of vehicle is out of the scope of this research, only the last measurement is important, which provides the total number of vehicles.

All locations except for 502 and 502r consist of two lanes. The traffic data is provided separately for each lane, and the lanes are numbered starting at the highway median. This implies that most vehicles are detected at the highest lane number, based on the assumption that most vehicles are inclined to drive on the rightmost lane. Moreover, both the start- and end timestamps are provided for each time interval. These are redundant, because the time interval is fixed, and it is chosen to remove the latter. In the end, the data set for each location to be processed includes the start date, start time, traffic flow, and lane number.

3-1-3 Data on factors potentially influencing traffic

The influence of external factors on traffic flow will be investigated. In addition, the applicable data will be utilized to improve the prediction model. The Koninklijk Nederlands Meteorologisch Instituut (KNMI) possesses 48 automatic weather stations in the Netherlands. The two weather stations adjacent to the area of interest are Ijmuiden and Wijk aan Zee. The wind direction, hourly average wind speed, average wind speed of the last 10 minutes, and the highest wind peak in the last hour are measured at Ijmuiden. The temperature, dew temperature, sun duration, global radiation, precipitation duration, hourly precipitation amount, and relative atmospheric humidity are measured at Wijk aan Zee.

During the implementation of the prediction model, weather forecasts will be used. However, only exact historical weather data is available, and the forecasts are not. Because the data is aggregated at one-hour time intervals, the weather data does not have to be very detailed. Therefore, it is assumed that the measured hourly weather specifications are similar to the weather forecasts. This implies that the measured data can be implemented during training without inducing a discrepancy in the data used during training and inference.

3-2 Data processing

The traffic data provided by the NDW has not been processed or validated. In the previous section, it was already indicated that the accuracy and availability of the sensors do not equal 100%. Therefore, the traffic flow data has to be investigated and cleaned to limit the amount of illogical data inputted into the model. On the other hand, the weather data provided by the KNMI is already validated. Therefore, it is assumed that further processing is not required.

3-2-1 Identification of invalid traffic data

To limit the amount of invalid data input to the model, the traffic data is investigated, and the following irregularities are found: misalignment's due to summer and winter time, erroneous and inaccurate traffic flow data, and missing data. To limit the effect of these irregularities, quality checks are implemented to locate these irregularities. Subsequently, modifications to the data set are performed.

Summer and winter time

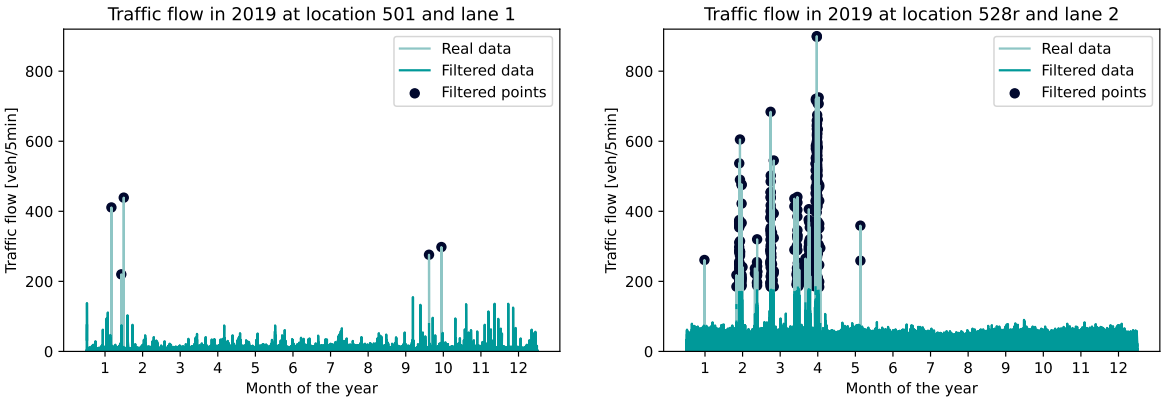
In the Netherlands, summer and winter time are applicable. This denotes that during the winter period, the Netherlands is in time zone UTC+1. On the last Sunday of March at 02:00:00, the time is advanced one hour, such that the time zone is UTC+2. The summer period lasts until the last Sunday of October, when the time is adjusted at 03:00:00. It is desired to match the traffic flow data to the summer and winter time zones, because daily trends, such as the morning rush hour, depend on the summer and winter time zone. In addition, continuous time series are required if models with a recurrent structure are implemented. The traffic data provided by the NDW is labeled by the summer and winter time [41]. To ensure that the data is continuous, an empty hour is inserted when summer starts. In addition, the last Sunday of October contains 25 hours of data. Therefore, the measurements corresponding to the same hour are averaged. Because these changes are applied when traffic is quiet, a very limited amount of information is lost.

Erroneous and inaccurate traffic flow data

To validate the data, different quality checks are performed. These checks should at least encompass erroneous and inaccurate data [57]. The first relates to implausible data, data that does not fall in an expected theoretical range, whereas the second relates to data that is inaccurate due to measurement errors, but falls within plausible ranges.

Two quality checks are performed to filter out the invalid data. First, the traffic flow is compared to a theoretical maximum to filter out erroneous data. Secondly, the distribution of measurements for each hour of the day is calculated, and the outer 5% is further investigated by looking into the surrounding data points, to perceive inaccurate data.

The theoretical maximum traffic flow $I_{\max}[\frac{veh}{5\ min}]$ is calculated as



(a) Filtered data in 2019 for location 501 and lane 1 **(b)** Filtered data in 2019 for location 528r and lane 2

Figure 3-2: Effect of quality check 1, implemented to filter out implausible data points in 2019 for location 501 lane 1 and for location 528r lane 2 in (a) and (b), respectively.

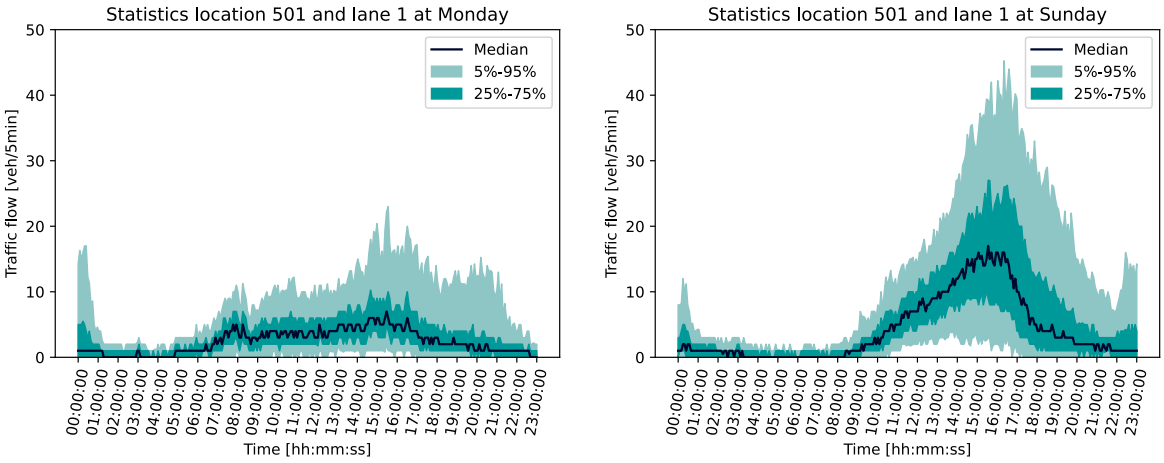
$$I_{\max} = \frac{5\,(v_{\max} \cdot f_{\mathrm{s}})}{s_{\mathrm{c}} + s_{\mathrm{dis}}}, \tag{3-1}$$

where $v_{\max}$ equals the maximum velocity in $\frac{m}{min}$, f_{s} is a safety factor to take fast driving into account and equals 1.3, s_{c} the average length of a vehicle assumed to be $4m$, and s_{dis} the minimum distance between two cars in m, calculated as

$$s_{\mathrm{dis}} = \frac{(v_{\max} \cdot f_{\mathrm{s}}) \cdot t_{\mathrm{dis}}}{60} \tag{3-2}$$

Here t_{dis} is equivalent to the following distance, set to $1.5s$. All these parameters have been rounded such that the theoretical maximum is as high as possible and only the actual infeasible points are filtered out. In reality, the maximum capacity depends on different circumstances, such as the weather. However, for simplicity, it is assumed to be constant. For each location, the theoretical maximum equals approximately $180\frac{veh}{5\ min}$ and if exceeded, the data point is removed from the data set. These theoretical maximum values are comparable to the values used in [62], which implements similar quality checks on arterial roads in the Netherlands. Therefore, the found values are assumed to be reasonable boundaries. To illustrate which data points are filtered, Figure 3-2 shows the real and filtered data for location 501 at lane 1 and location 528r at lane 2 in 2019, in Figure 3-2a and 3-2b, respectively. These clearly show that only the high, implausible peaks are filtered out. The high consecutive peaks of filtered data points for location 528r are significant. In Section 3-1-1 the importance of the sensitivity of the electromagnetic loop detector was discussed. The behavior shown for location 528r might be because the detector is too sensitive. Moreover, the solitary peaks might be caused by a stationary vehicle, which is repeatedly detected in a short amount of time.

The first quality check has removed the implausible data points from the data set. However, measurement errors also exist in the plausible ranges, as indicated in [47]. Traffic flow has a clear trend over the hours of the day for each day of the week (dow). To detect the outliers, the median and $5\% - 95\%$ percentiles are calculated for each hour of each dow. In Figure 3-3, these statistics are shown for location 501 and lane 1, for Monday and Sunday. These



(a) Statistics Monday

(b) Statistics Sunday

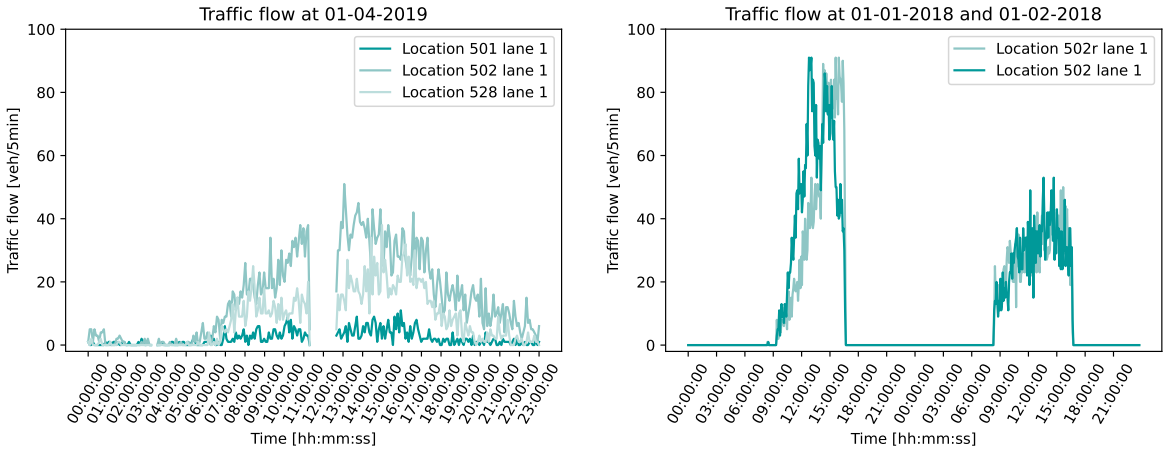
Figure 3-3: Median and percentile statistics for location 501 and lane 1 at Monday and Sunday

figures clearly illustrate the difference in traffic flow behavior imposed by the dow and clarify the decision to investigate each dow separately. The data points outside these boundaries are further investigated by looking into the surrounding data points representing the traffic flow of 15 minutes before and after. This is based on the intuition that if extraordinary behavior occurs, this will also be indicated by consecutive measurements. If irregular behavior solely occurs at one data point, it is assumed to be a measurement error, and removed from the data set.

In practice, whether a data point is labeled as inaccurate depends on two conditions. First, the traffic flow has to be more than three times the value of the median of the previous and succeeding 15 minutes. However, it is observed that by solely applying this condition, abundant traffic flow data at quiet hours are filtered out. On the contrary, it is undesired to enlarge the threshold of three times the median, because that allows strange peaks to go unnoticed. Therefore, a second condition is applied, which constraints the inaccurate data to be at least $50\frac{veh}{5\ min}$. The data set based on a 5 minute interval will be further aggregated to hourly time intervals. Therefore, it is assumed that small unnoticed measurement errors are not that significant in the final data set. In addition, the objective is to model the irregular traffic flow behavior caused by the weather or events. Therefore, it is undesired to filter irregular data points that might be caused by these factors instead of measurement errors. For locations 528 and 528r, most data points are removed. However, this is still insignificant compared to the entire data set. As an illustration, the percentage of total data points removed for these locations equals approximately 0.5% and 0.7%.

Missing data

Besides the data points removed from the data set due to the quality checks in the previous section, the data has a few missing values. For each location and lane, a total of approximately 1000 measurements are missing throughout the entire set, which accounts for a total of 0.3% of the entire data set. These defects occur at the same time for adjacent links, and often for



(a) Missing data at 01-04-2019 **(b)** Zero data at 01-01-2018 and 01-02-2018

Figure 3-4: Illustration of two types of implausible data in the data set.

multiple detectors as well. Therefore, they are assumed to be caused by a defect in the NDW system and account for the unavailability of the system described by Polman. An example of this behavior is shown in Figure 3-4a, which shows the traffic flow data at 01-04-2019 for three locations.

In addition, it is observed that for certain locations, the traffic flow equals zero for a period of over 6 hours, as shown in Figure 3-4b. Since this is implausible for a period this long, these measurements are removed from the data set and taken into account in the next section. This behavior seems planned, since either a consecutive number of days only have zero traffic flow measurements, or several consecutive days have zero traffic flow measurements at a fixed interval. Therefore, this behavior might be caused by planned construction works.

3-2-2 Response to missing or implausible data

Different methods are implemented in state-of-the-art literature to cope with missing values in the data set, some authors use linear interpolation [22, 36, 66], others insert the average of neighboring data points [72], and [35] even filters out an entire month because it contains several missing values. However, in this research, the data still has to be aggregated into larger time intervals. Therefore, if less than four consecutive data points (20 minutes) are missing, the data is assumed to still represent the hourly traffic behavior, and the number of missing values is taken into account during aggregation.

It is undesired to have missing values during the day because models incorporating auto-correlation are not able to anticipate this. To ensure that all the prediction models are subject to the same data set, it is chosen to remove the days that contain more than 3 consecutive missing values. However, the statistics shown in Figure 3-3 indicate that between 01:00:00 and 04:00:00 traffic flow is approximately zero with a very small variance. It would be a shame to miss an entire day of data based on missing values in this time range. Therefore, if the missing values lie in this time range, the median of the corresponding hour and day is inserted, and the remaining data is kept.

The total number of removed days per location and lane lies in the range of 30 and 50 days, which is equivalent to 2.74% to 4.57% of the total data set. The only exceptions are location 530 at lane 1 and location 530r at lane 1, for which a total of 183 and 249 are removed from the data set, respectively. This is mainly caused by the unavailability of the detectors in the first six months of 2017. The processed data is still an adequate representation of the entire data set because the removed days are divided over the years, seasons, and days of the week.

3-3 Data preparation

After the preprocessing steps, the data is prepared such that it can be implemented in the correlation analyses and the prediction models. This is done in three steps: data aggregation, the addition of different lanes, and standardization of the data set. At last, the median of the weekly traffic flow is investigated for all locations, to make a well-thought consideration on which locations there should be elaborated while maintaining genericity.

3-3-1 Data aggregation

First, the data is aggregated into one-hour time intervals, because the objective is to make an hourly prediction. The aggregated traffic flow ($I_{\text{day,hour}}$)is calculated as

$$I_{\text{day,hour}} = \sum_{i=1}^{12} I_{\text{day,hour},i} \cdot \alpha_{\text{day,hour}}, \tag{3-3}$$

in which the sum is taken of the traffic flow measurements at a five-minute interval ($I_{\text{day,hour},i}$) of the corresponding day and hour. In addition, $\alpha_{\text{day,hour}}$ accounts for the remaining missing values in an hour and is calculated as

$$\alpha_{\text{day,hour}} = 1 + \frac{x_{\text{day, hour}}}{12}, \tag{3-4}$$

where $x_{\text{day,hour}}$ equals the number of missing values at the corresponding day and hour.

3-3-2 Adding the data of multiple lanes

In the sections above, the traffic flow data of different lanes for the same location has been investigated separately. However, it is not desired to predict the traffic flow of the two lanes separately, because the final objective is to predict the total traffic flow and the two lanes will be highly correlated. Therefore, it is inconvenient to predict both traffic flows separately and add these afterward. Hence, the traffic flow data of both lanes are added. Because the days removed from the data set are not necessarily equal for both lanes, it is chosen to remove the specific days from both data sets. For simplicity, it is chosen to look at the different lanes simultaneously, even though a small amount of relevant data will be lost.

3-3-3 Feature scaling

At last, it is important to scale the input features, such that these lie in a comparable range, and one input feature does not impact the model solely because its value is relatively high. In addition, many optimization algorithms converge faster after scaling. As an illustration, the gradient descent algorithm descends quickly on a steep slope and slowly on a horizontal plane. If the variables are of incomparable size, the algorithm will oscillate inefficiently and slowly converge to the optimum.

There are two main feature scaling techniques, often implemented in the state-of-the-art literature, min-max normalization [22] and z-score standardization [36, 66, 67]. The scaled feature x_s is calculated from the original input x by min-max normalization as

$$x_s = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{3-5}$$

where $x_{\max}$ is equivalent to the highest and $x_{\min}$ to the lowest value in the feature set. Consequently, the scaled features lie in a range of 0 and 1. A disadvantage of this scaling method is that it is very sensitive to outliers. As an illustration, if a high outlier is present, after scaling, the outlier will be close to 1. However, all the other values will be low and close to each other, since they are scaled with $x_{\max}$, which equals the high outlier value.

Z-score standardization assumes the data to be normally distributed and uses the mean (μ) and the standard deviation (σ), to scale the data between -1 and 1. The scaled feature x_s is calculated as

$$x_s = \frac{x - \mu}{\sigma} \tag{3-6}$$

This method is less prone to outliers and is used more often in the state-of-the-art literature. Therefore, it is chosen to use z-score standardization for feature scaling.

3-3-4 Comparison of traffic behavior for multiple locations

Figure 3-5, shows the weekly median of the hourly traffic flow for all locations. This indicates that the locations can be divided into two groups: locations 501, 501r, 502, and 502r, and the other locations. The first group experiences less traffic flow during the week, and an increase in flow during the weekend, whereas, the other locations experience the opposite. Because the behavior in a group is very similar, it is assumed that investigating one location in each group is sufficient to investigate the genericity of the prediction model. Therefore, in the remainder of this research, the focus lies on locations 501 and 531.

3-4 Summary

This chapter discussed the available data in the Netherlands, the case study, and the processing steps undertaken to limit the amount of invalid data. For the scope of this research, it is chosen to focus on data, bound to the infrastructure. More specifically, it is chosen to use traffic data provided by the NDW, due to the large historical availability. However, if desired, this can be extended by using other data sources later on. In addition, weather data

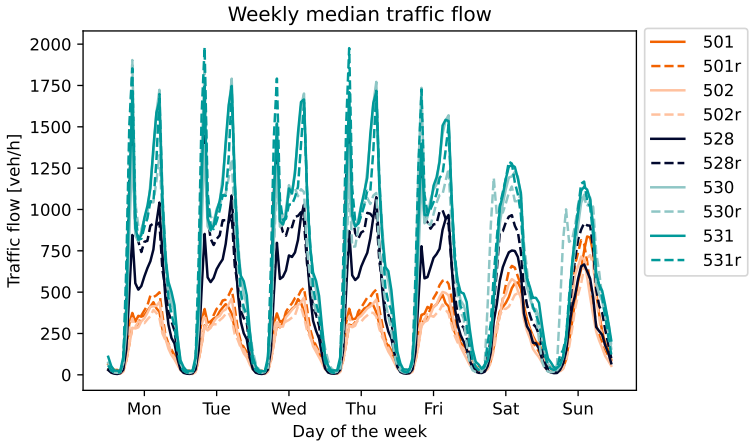


Figure 3-5: Weekly median traffic flow for all locations.

is provided by the KNMI, which is already validated, and consequently does not have to be processed anymore.

The case study investigated throughout this research is the area of Haarlem, Zandvoort, and Bloemendaal aan Zee. This specific location is chosen because it is known to be highly influenced by external features. Traffic flow data, aggregated at 5 minutes, of 10 locations are available, for 2017, 2018, and 2019. It is chosen not to include 2020 because the traffic behavior is significantly different from the other years due to the COVID-19 pandemic. Therefore, it is assumed that this data is not representative of the impending years.

The provided traffic data has not been processed or validated. Moreover, the properties of the detectors indicate that the accuracy equals approximately 95-98% and the system is available 97-100% of the time. This shows the necessity to process the data, to limit the amount of invalid data input to the model. First, the misalignment due to summer and winter time is modified. Next, two quality checks are performed to identify erroneous and inaccurate traffic data. The first check compares the traffic flow with the theoretical maximum, to remove implausible data points. The second check, looks into measurement errors inside the plausible ranges, by comparing irregular data points with their neighbors. At last, missing data and implausible zero data are identified.

Days are removed from the data set when two requirements are met. First, more than three consecutive data points are missing, which lie outside 01:00:00-04:00:00. This is based on the assumption that if fewer data is missing, the corresponding data still represents the hourly traffic behavior. Moreover, at the chosen time interval, there is shown to be little to no traffic flow. In the end, 2.74% to 4.57% of the data set is removed, depending on the location.

Next, the data is prepared in three steps such that it can be implemented in the correlation analyses and the prediction models. The data is aggregated into an hourly interval. Next, the data of multiple lanes for the same location is added. At last, z-score feature standardization is implemented to ensure that multiple features are in the same range.

Finally, the traffic behavior of multiple locations is compared by investigating the weekly median. To test the genericity of the models, it is chosen to focus on locations 501 and 531 in the remainder of this research, because these are subject to different traffic behavior.

Chapter 4

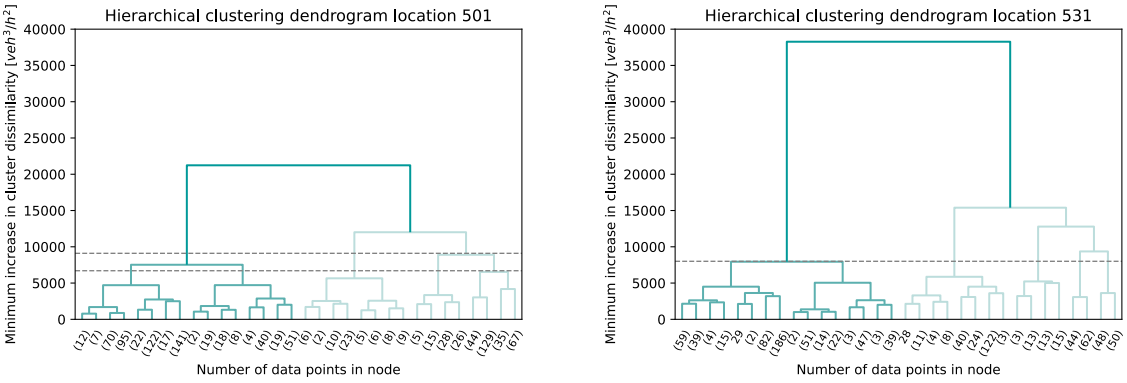
Analysis of auto- and cross-correlations in traffic flow and external features

This chapter investigates the correlations in the data set, such that a hypothesis can be made on which features are important in traffic flow. This is done in three steps. First, clustering is performed on the different days in the data set, to investigate the similarity between days and whether this can be assigned to intuitive features. Next, cross-correlation analyses are implemented between traffic flow and weather features, to identify important and redundant features. At last, the auto-correlation in traffic flow data is investigated, to indicate the importance of previous traffic flow measurements on different prediction horizons.

4-1 Daily clustering

Agglomerative hierarchical clustering is implemented as described in Section 2-1-1, with the publicly available *sklearn.cluster.AgglomerativeClustering* library. To look into the similarities and differences between days, first, the data set is transformed into features corresponding to traffic flow during one day. As a result, each day is represented by 24 values. Next, clustering is performed to find an indication of the similarity between days. The hourly features are not scaled, because by implementing scaled features, it was found more difficult to identify irregular days. This is explained by the fact that hours with fewer traffic flow will now contribute equally to the similarity score. Moreover, similar intuitive features were found to be assigned to the distribution. Therefore, it is chosen to show the clustering based on the regular traffic flow in this section.

The important defined parameters are the affinity, equal to the Euclidean distance, the linkage, set to Ward, and the distance threshold, which is set to *None*. The first two parameters represent the metrics described in Section 2-1-1. In addition, by setting the distance threshold to *None*, the entire dendrogram is computed, by continuing until all days are merged. Based on the dendrogram a rational number of clusters is chosen and the corresponding clusters are investigated.



(a) Dendrogram location 501 **(b)** Dendrogram location 531

Figure 4-1: Dendrograms for locations 501 and 531 obtained by hierarchical clustering in (a) and (b), respectively. The dotted lines represent the intuitive division of the data set in 3 and 5 clusters for location 501, and 6 clusters for location 531.

4-1-1 Dendrogram

To acquire insights into the number of clusters in the data set, the dendrograms retrieved for locations 501 and 531 are shown in Figure 4-1a and 4-1b, respectively. The first dendrogram indicates that based on the minimum increase in cluster dissimilarity, two logical choices are to divide the data set into three or five clusters. The corresponding horizontal slicings are indicated by the gray dotted lines at approximately $9000 \frac{veh^3}{h^2}$ and $6000 \ \frac{veh^3}{h^2}$. Moreover, the second dendrogram indicates that an intuitive division can be made at six clusters for location 531 at $6700 \ \frac{veh^3}{h^2}$. These clusters are further investigated in the next section.

For comprehensibility, it is chosen to show the top 4 layers of the dendrogram. The values shown on the x-axis indicate the number of days inside the corresponding cluster. The clusters obtained for a lower increase in cluster dissimilarity are also investigated. However, these result in several clusters containing only a few irregular days, to which no intuitive features can be assigned. This is illustrated in Appendix A-2, which shows the clusters for location 501 when divided into seven clusters. Moreover, this is also indicated by the values on the x-axis of the dendrogram, which show many clusters with a limited number of days. Therefore, to find relevant features, the division into more clusters is not further investigated.

4-1-2 Clustering results

Which days correspond to the same cluster is investigated for the options obtained in the previous section. First, location 501 is investigated for three clusters. Figure 4-2 shows the clusters obtained in a calendar plot, indicating the days of the year belonging to each cluster. In addition, the right figure represents the corresponding cluster centers, where the corresponding days and cluster centers are indicated by the same color. In addition, the white days correspond to days that have been filtered out in the data analysis.

From the clusters, it can be concluded that there is a clear difference in traffic flow behavior during summer and winter, indicated by clusters 1 and 2, respectively. Therefore, the season is an important feature to describe traffic flow behavior. In addition, each summer contains

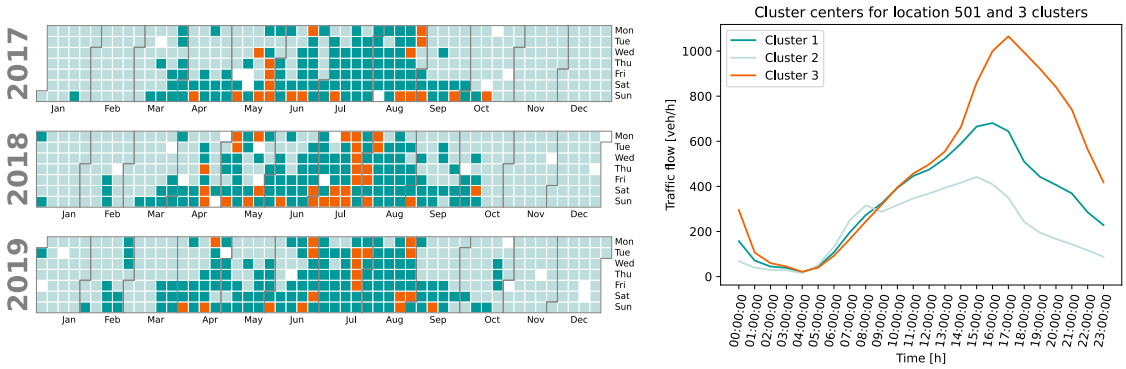


Figure 4-2: Specifications of the 3 clusters obtained by hierarchical clustering for location 501. The left figure shows the division of the days corresponding to each cluster throughout the year, and in the right figure the corresponding cluster centers are shown.

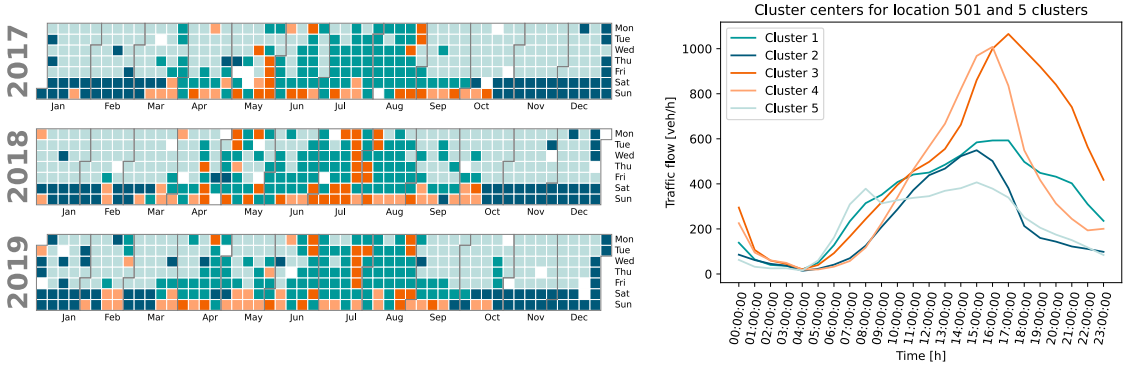


Figure 4-3: Specifications of the 5 clusters obtained by hierarchical clustering for location 501. The left figure shows the division of the days corresponding to each cluster throughout the year, and in the right figure the corresponding cluster centers are shown.

a few irregular days clustered into cluster 3, which are shown to have significantly more traffic flow than the days corresponding to the other clusters. Since location 501 lies on the road towards the beach, and days corresponding to cluster 3 only appear in summer, it is hypothesized that the increased traffic flow might be caused by good weather.

Next, the data set is clustered into five clusters, again the days corresponding to each cluster are shown in a calendar plot, together with the cluster centers in Figure 4-3. Interesting to see is that a clearer division between weekdays and the weekend is present. This indicates that the day of the week (dow) will also be an important input feature for the prediction model.

Subsequently, the same analysis is done for location 531. The clusters for a division of the days into six clusters are shown in Figure 4-4. Interesting to see is that different behavior is noticed compared to location 501. This location seems to be highly influenced by the school vacations. During these vacations, the pattern of the traffic flow is equivalent to that on regular days, but the amount is reduced. In addition, Friday, Saturday, and Sunday each

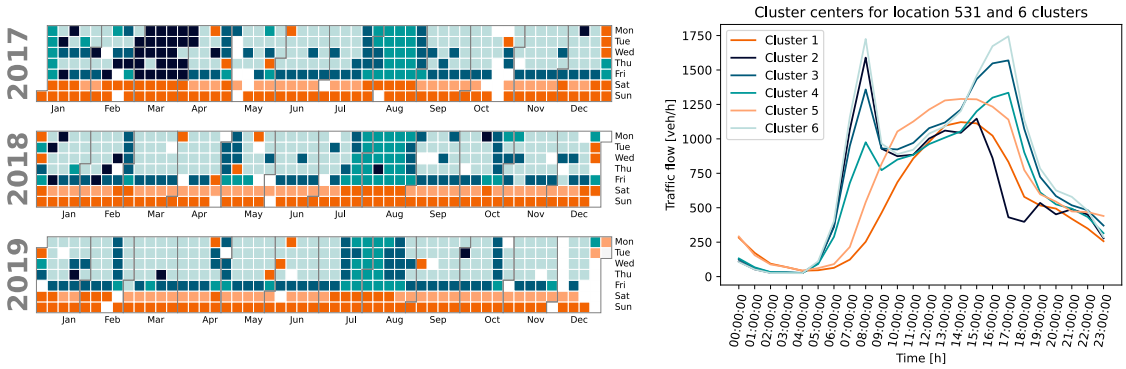


Figure 4-4: Specifications of the 6 clusters obtained by hierarchical clustering for location 531. The left figure shows the division of the days corresponding to each cluster throughout the year, and in the right figure the corresponding cluster centers are shown.

have their specific behavior. Moreover, both locations experience a different traffic behavior on national holidays, such as the first of January. Location 501 experiences very irregular behavior, whereas for location 531 it is comparable to a Sunday.

Therefore, it is concluded that the dow, season, school vacations, and national holidays are important features influencing traffic flow behavior, which should be taken into account in the prediction model. Even though some of these features are more important for one location than the other, for genericity it is chosen to take all features into account for both locations.

4-2 Weather and traffic flow cross-correlation

To identify important and redundant weather features, cross-correlation analyses are performed. Both Pearson's and Spearman's rank methods are implemented, where the latter was able to find higher correlations. Therefore, in line with the research done in [31], the second method is preferred and chosen to be elaborated on. Moreover, by plotting the data, the assumption regarding the monotonic relation appears to hold.

Spearman's rank is applied to the traffic flow and available weather variables. The correlation matrix found for location 501 is shown in Figure 4-5, in which dark green means a strong positive correlation, dark orange a strong negative correlation, and white no correlation. The correlation matrix indicates that the temperature, dew temperature, sun duration, and radiation are positively correlated to the traffic flow. In addition, the relative humidity is negatively correlated to the traffic flow.

The precipitation does not seem correlated with the traffic flow. This is unexpected, both intuitively and based on state-of-the-art literature [31, 73]. This might be caused by the discrepancy in the location of the road sensors and weather stations. Where this does not seem to be an issue for the other features, precipitation is relatively more location-specific. Therefore, the precipitation features might contain errors regarding the actual precipitation.

Redundant features are identified as features, excluding the traffic flow, that are highly correlated to each other. These features do not both have to be input into the prediction model,

because they contain similar information. The correlation matrix indicates that the temperature and dew temperature, the sun duration and radiation, the precipitation duration and precipitation sum, and at last the wind features are redundant.

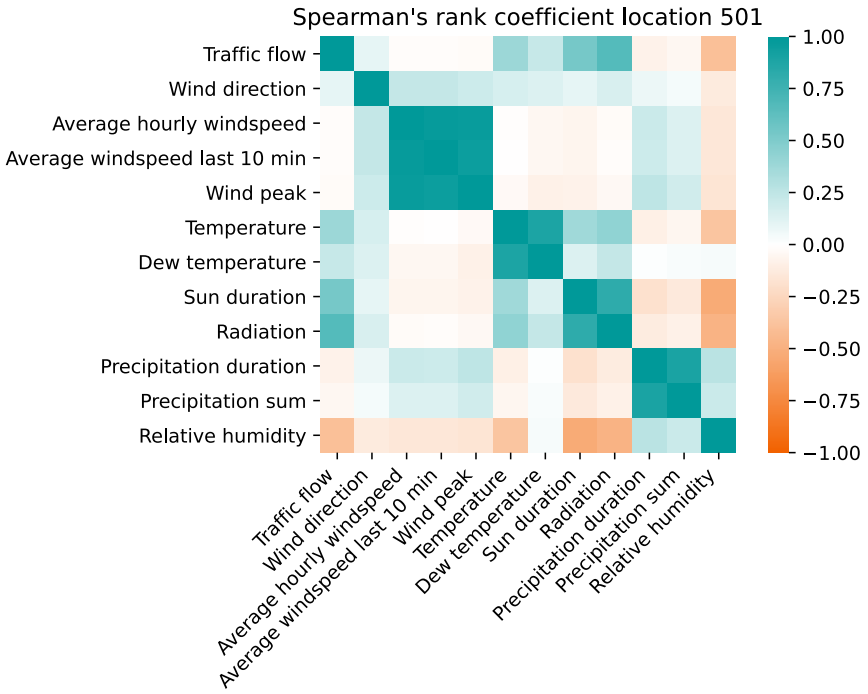


Figure 4-5: Spearman's rank correlation coefficient for location 501.

The cross-correlation coefficients are also investigated for the other locations and are found to be similar. The only difference in correlation coefficients is noticed in the temperature and dew temperature. These are found to be more important for locations 501, 501r, 502, and 502r, as opposed to the other locations. The difference between these locations is that the first lie on the road towards the coast, whereas the others lie on the ring road of Haarlem. The last is more subject to periodic commuter traffic, and the first to irregular traffic, which explains the difference in correlations.

The previous section has indicated that the traffic behavior is influenced by the time of the day and the dow. Therefore, additional analyses are done to investigate whether the weather features are more or less correlated at a specific hour of the day or dow. The first analysis is performed by subtracting the median traffic flow from the data set. In addition, a second analysis is done by investigating the correlations for each dow separately. However, the same features were found to be important.

To conclude, the weather features that are correlated to the traffic flow and should be input to the prediction models for these specific locations are the temperature, radiation, and relative humidity. Moreover, it is chosen to neglect the dew temperature and sun duration due to the redundancy. In addition, these have a slightly lower correlation coefficient with traffic flow than the temperature and radiation.

4-3 Auto-correlation in traffic flow

The auto-correlation in traffic flow is investigated to explore the historical traffic flow measurements that should be taken into account on different prediction horizons. In short-term predictions, these auto-correlations are already shown to be important [36, 67, 74]. However, the prediction horizon up to which these correlations are significant still has to be investigated. Therefore, the traffic flow is shifted 1 to 504 times, to obtain 505 different vectors, such that a maximum correlation of three weeks ago can be investigated. The limit is set to three weeks, due to computational constraints. Next, Pearson's and Spearman's rank correlation coefficients are calculated.

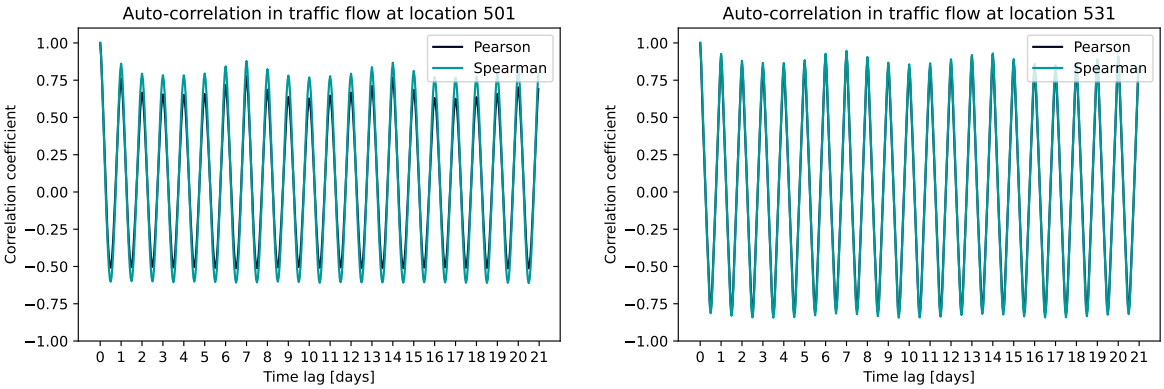
Figure 4-6a and 4-6b illustrate the auto-correlation for location 501 and 531, respectively. These graphs indicate a clear correlation with time because the current traffic flow is comparable to the traffic flows at a similar time the previous days. Moreover, again the dow seems important, because a higher correlation is observed at the same dow.

In line with [31], the data is detrended to remove fluctuations caused by the hour of the day and dow. As a result, it is easier to see whether correlations are caused by other factors. Therefore, to investigate, the correlations without the influence of the time of the day, the hourly median of the traffic flow is subtracted from the data. To avoid any misconceptions, this is similar to standardizing the traffic flow based on the hour of the day, because both correlation methods are scale-invariant. The difference is that standardization takes the mean into account instead of the median. Figure 4-6c and 4-6d show the corresponding correlation coefficients. For both locations, the traffic flow of the past few hours, and of a similar time the previous day, are important. In addition, the same dow and neighboring days are also shown to be important. This indicates the importance of the dow.

At last, to decrease the influence of the dow, the weekly median is subtracted from the original traffic flow. The corresponding correlations are shown in Figure 4-6e and 4-6f. A clear difference in traffic behavior between the two locations is shown in these figures. Location 501 correlates with all previous days, at approximately the same hour of the day. Moreover, a slightly higher correlation is noticed for the same dow a week ago. On the contrary, location 531 still indicates a high correlation with the same and neighboring days at a similar time one, two, and three weeks ago. However, no correlation is found with the other days. Moreover, the correlation pattern for both locations decreases as the prediction horizon increases.

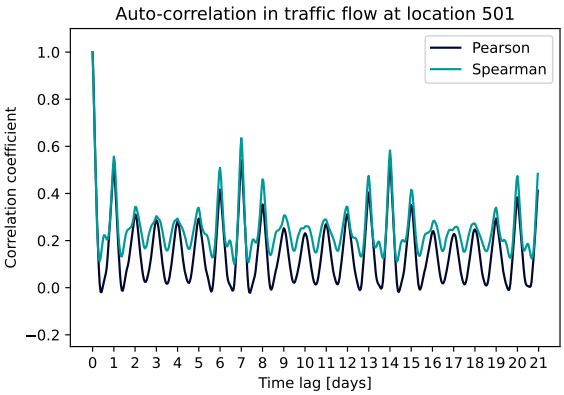
These analyses have also been implemented for the other locations, and a similar behavior was found for the locations categorized in the same group. The difference in behavior between the two groups might be because they are subject to different trends. The clustering analyses in Section 4-1 showed the importance of the season and dow for location 501 and 531, respectively. Therefore, it is reasonable that the first is more correlated to all previous days, and the second is correlated to similar days of the week.

To conclude, the current traffic flow is important for a prediction horizon up to approximately 4 hours. If the prediction horizon increases further, the correlation starts to decrease and is not significant anymore. However, when the horizon increases further, it becomes significant again when the time of the day is similar again. For computational constraints, it is chosen to limit the number of previous measurements to be taken into account to 48 hours. Location 531 shows a clear correlation with the same dow a week before. Therefore, it can be beneficial to extend the input data with the traffic flow at the same hour a week ago, as done in [5, 22, 24].

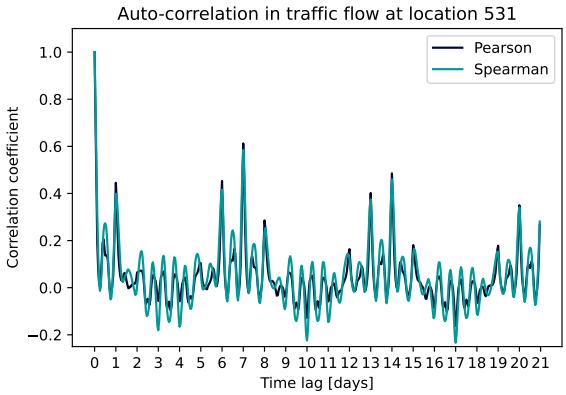


(a) Location 501

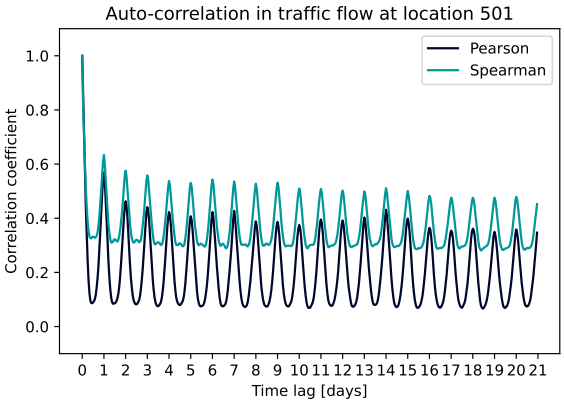
(b) Location 531



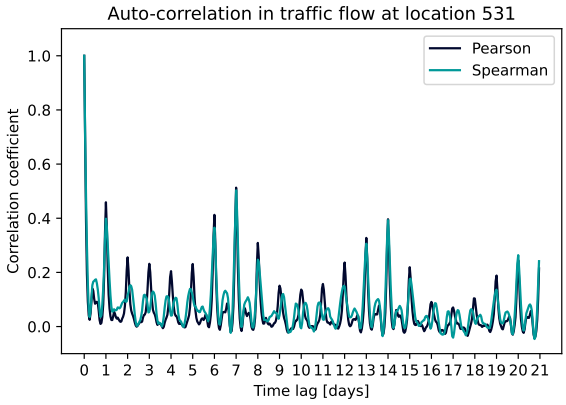
(c) Location 501



(d) Location 531



(e) Location 501



(f) Location 531

Figure 4-6: Auto-correlation of the hourly traffic flow for 3 consecutive weeks. Where (a) and (b) look into the true traffic flow, (c) and (d) into the traffic flow without the influence of the daily median, and (e) and (f) into the traffic flow without the influence of the weekly median.

4-4 Summary

This chapter has investigated the correlations in the data set, to gain insights into the importance of different features on traffic flow, by implementing multiple correlation analyses.

First, daily clustering is performed by implementing agglomerative hierarchical clustering. Based on the dendrograms, a clear split into 3 and 5 clusters was observed for location 501, and a split into 6 clusters for location 531. By examining these clusters, the dow, season, school vacation, and national holidays were found to be important intuitive features. The importance of the features differs per location. However, for the genericity of the prediction model, it is chosen to input all features into each model.

Next, cross-correlation analyses are implemented between the available weather features and the traffic flow. Based on Spearman's rank method, temperature, radiation, and relative humidity are found to be important weather features. The same correlations were found for the different locations, except for the temperature and dew temperature, which were more significant for location 501. In addition, even though a significant correlation was found, the dew temperature and sun duration can be neglected, due to redundancy.

At last, the auto-correlation in traffic flow is investigated, to identify the previous traffic flow that should be taken into account on different prediction horizons. To decrease the influence of the time of the day and the dow, additional analyses are done on the traffic flow without the influence of the daily and weekly median. In the end, the traffic flow of the past few hours is shown to be correlated to the current traffic flow. In addition, traffic flow at a similar time the previous day is correlated as well. Moreover, location 501 is shown to be correlated to all previous days at a similar time, whereas location 531 only shows high correlations for the same and neighboring dow at a similar time. This indicates that the current traffic flow is correlated to traffic flow on further horizons, which implies that the transformer might be beneficial on longer prediction horizons. Due to computational constraints, it is chosen to limit the past input features to implement to the last 48 hours. However, it might be beneficial to extend the input data with the traffic flow at a similar time a week ago.

These differences between the locations are explained by the location characteristics. Location 531 is mainly affected by commuter traffic, and location 501 has more irregular behavior. Due to the commuting traffic, the traffic flow is expected to decrease during school vacations and show a strong similarity between the dow, which furthermore supports that in the auto-correlation similar dows have a stronger correlation. On the other hand, it is reasonable that the road to the coast is more influenced by the season, which is related to the temperature. This additionally supports the auto-correlation with all previous days.

To summarize, the dow, season, school vacation, national holidays, temperature, radiation, and relative humidity are features that should be input into the model. In addition, the influence of auto-correlation seems applicable to longer prediction horizons.

Chapter 5

Baseline models for long-term traffic flow prediction

In Chapter 4 important features to describe traffic flow behavior have been identified. How to implement and convert these features into the final feature set will be described in Section 5-1. Next, the baseline models, the random forest and multilayer perceptron (MLP) are constructed in Section 5-2, based on Bayesian hyperparameter optimization and an evaluation on the validation set. At the end of this chapter, the final baseline models are obtained, such that they can be evaluated and compared to the transformer in Chapter 7.

5-1 Configuration of the input feature set

In the correlation analyses in Chapter 4, the hour of the day, day of the week (dow), season, school vacations, national holidays, temperature, radiation, and relative humidity were found to be important features that should be taken into account in the prediction model. The implementation of these features is discussed. Next, the final data set is described.

5-1-1 Conversion of information to input features

The important features are categorized. First, the hour of the day, season, and dow are grouped into the temporal periodic features. In addition, the national holidays and vacations belong to the temporal categorical features. At last, the weather features are grouped. The implementation of these features is discussed in the next subsections.

Temporal periodic features

The hour of the day can be represented by 24 different values (h) ranging between 1 and 24. The advantage of this method is that only one feature is required to represent time. However,

deep learning models assume that values close to each other have a higher correlation than values further apart. As an illustration, 1 has a higher correlation to 2 than to 18. In reality, this also holds for time. However, since time is cyclical, a strange jump occurs between 24 and 1. Therefore, another option is to represent time by a cosine and sine feature. As a result, the time is cyclical and each hour of the day is represented by a unique feature. The integer time values (h) are converted to $\text{time}_{\cos,h}$ and $\text{time}_{\sin,h}$ as:

$$\text{time}_{\cos,h} = \cos\left(\frac{2\pi h}{24}\right), \quad \text{time}_{\sin,h} = \sin\left(\frac{2\pi h}{24}\right) \tag{5-1}$$

Despite the increase from 1 to 2 required features to represent time, this method is implemented, because it better represents the cyclical behavior.

From the clusters obtained in the correlation analyses, three observations are made regarding the dow. First, different traffic behavior is observed on different days of the week. In addition, the similarity between days differs per location. At last, consecutive days are not necessarily more similar. Therefore, to maintain the genericity of the model and to ensure that no false assumptions are imposed on the model, the dow is treated as a categorical feature and converted by one-hot-encoding [20]. As a result, the dow is represented by 7 categorical features, which are equal to 1 when the data point corresponds to the respective dow, and 0 otherwise.

The same characteristics as for the time of the day hold for the season. Therefore, similarly, the season is represented by two features $\text{season}_{\cos,h}$ and $\text{season}_{\sin,h}$. In addition, when implemented as a categorical feature, it is required to make a hard split between seasons, which will be location-dependent. On the other hand, a cyclical feature better represents the transition between seasons. In addition, if the season is implemented as two cyclical features, indirectly not only information about the season, but also about the day and month of the year is provided.

Temporal categorical features

School vacations and national holidays are shown to be subject to irregular traffic behavior. It is chosen to implement these as two separate categorical features, vac_t and free_t, respectively. Because the traffic behavior is different during school vacations and national holidays, these are not implemented as a single feature. Moreover, different national holidays are also subject to different behavior. However, dividing these days into even more features, causes the data corresponding to the features to be even more sparse. Therefore, it is chosen to include these in the same feature.

Weather features

In the cross-correlation analyses, it was determined that temperature, radiation, and relative humidity are important weather features. In Section 3-3-3, the importance of feature scaling has been discussed. Therefore, these weather features are standardized and inputted into the model, separately. These features are referred to as temp_t, hum_t, and rad_t.

Based on the feature implementation described above, the final input feature $x \in \mathbb{R}^{16}$ corresponding to time t, is composed as

$$x_t = \left[\text{time}_{\text{sin},t}, \text{time}_{\text{cos},t}, \text{dow}_t, \text{season}_{\text{sin},t}, \text{season}_{\text{cos},t}, \text{free}_t, \text{vac}_t, \text{temp}_t, \text{hum}_t, \text{rad}_t\right], \quad (5\text{-}2)$$

where

$$\text{dow}_t = \left[\text{mon}_t, \text{tue}_t, \text{wed}_t, \text{thu}_t, \text{fri}_t, \text{sat}_t, \text{sun}_t\right] \quad (5\text{-}3)$$

5-1-2 Final data set for training and inference

Some research divides the data set and trains multiple models, each specialized for a specific part of the data set. For example, in [31], two prediction models are trained, one for weekdays and the other for days during the weekend, causing each prediction model to be specialized on the corresponding data. The cluster analysis in Section 4-1 indicated that for this research this division can also be made because these days are alike. However, additionally, the cluster analysis indicated that days subject to irregular behavior occur on different days of the week. One of the objectives of this research is to find external factors, that might have caused this irregular behavior, and to take these into account in the prediction model. Therefore, dividing the data set to obtain multiple prediction models is not investigated, because the data containing irregular traffic behavior is already sparse and will degenerate even further if multiple models are implemented, making it more difficult to be captured.

The correlation analyses in the previous chapter indicated that the traffic behavior differs per season. Therefore, to be able to investigate the performance of the prediction model on the different days throughout the years, the first two years are taken as the training set and the last year as the test set. It is chosen not to randomly divide the data set, because the transformer model, implemented in the next chapter, is based on the previous traffic flow and multistep predictions. Therefore, if a random data set would be taken as the training set, indirectly the transformer would already be subject to the data in the test set. To fairly evaluate the transformer, it is desired to keep the train and test set completely separated. In addition, the same test and train data should be used in the baseline models and the transformer, to be able to make a fair comparison.

The validation set is often taken as 10-20% of the total data set [5, 24]. Therefore, it is chosen to allocate 20% of the training set to the validation data, which equals 13% of the total data set. As opposed to the test set, taking the last 20% of the training set as the validation set will not give a representative data set of the entire year. Moreover, evaluations based on this division indicated that the prediction models were always underfitting. This is expected because the correlation analyses in Chapter 4 showed that most irregular behavior occurs during summer. Therefore, no conclusions can be drawn based on this validation set. On the other hand, as indicated above, by randomly selecting the data, the transformer might indirectly already be subject to the unseen data. However, because no conclusions can be drawn regarding the first option, this split is assumed to be better suited.

To summarize, the first two years of the data set are taken as the training set, of which a random 20% is set aside as the validation set. Moreover, the last year is taken as the test set.

Table 5-1: Parameter space for Bayesian hyperparameter optimization of the random forest.

	$n_{\text{estimators}}$	Max depth	Min samples leaf
Parameter space	$100, 200, \ldots, 600$	1,2, ... , 300	1,2, ... , 30

5-2 Implementation of the baseline models

Two baseline models, that do not take auto-correlation into account, are the random forest and MLP, which are implemented in Section 5-2-1 and 5-2-2, respectively. Moreover, the code used to build, train, and evaluate these models is available at `https://github.com/carmenpetsch/Transformer.git`.

5-2-1 Random forest

The publicly available `sklearn.Ensemble.RandomForestRegressor` library [46], is used to implement the random forest, based on the input features as stated above.

Set-up of the random forest

Bayesian hyperparameter optimization is performed on the parameter space defined in Table 5-1. Where $n_{\text{estimators}}$, max depth, and min samples leaf denote the number of decision trees, the maximum depth, and the minimum number of samples at the end of each branch. The optimal parameters found after 100 evaluations are given in the first row of Table 5-2 for location 501. Next, the performance on the train and validation set is evaluated through the root mean squared error (RMSE). In addition, the decrease in performance (δ_{rf}) from the train to the validation set is then calculated by the relative increase in the RMSE. The corresponding performance measures are also given in Table 5-2.

The substantial δ_{rf} value indicates that the model is overfitting on the data. Therefore, the model hyperparameters are tuned. To decrease the effect of overfitting, $n_{\text{estimators}}$ and the minimum samples per leaf can be increased, whereas the maximum depth should be decreased. By manually tuning these hyperparameters, it was found that increasing $\text{n}_{\text{estimators}}$ has little effect. On the other hand, by decreasing the maximum depth and increasing the minimum samples per leaf the relative increase was shown to decrease significantly, while only slightly decreasing the performance on the validation set. Therefore, the parameters in the second row of Table 5-2 are implemented in the final random forest for location 501. Moreover, the corresponding performance measures are written in bold.

A similar optimization and evaluation are performed for location 531. The optimized and final hyperparameters are given in the last two rows of Table 5-2. Similar behavior was noticed; the optimized hyperparameters are equivalent, and the model experienced overfitting. However, to decrease δ_{rf}, the model had to be constrained a bit more. By changing the hyperparameters as described above, the random forest is made less specific. The correlation analyses indicated that location 531 is subject to less irregular traffic flow because it is mainly composed of commuter traffic. Therefore, this supports the explanation that a relatively simple model is

Table 5-2: Hyperparameters and performance measures of the random forest for locations 501 and 531.

Location	$n_{\text{estimators}}$	Max depth	Min samples leaf	$\text{RMSE}_{\text{rf,train}}$ $[\frac{\text{veh}}{\text{h}}]$	$\text{RMSE}_{\text{rf,val}}$ $[\frac{\text{veh}}{\text{h}}]$	δ_{rf} [%]
501	400	16	1	33.31	78.04	134.21
501	400	10	8	**74.67**	**83.11**	**11.30**
531	300	19	1	48.38	139.28	187.90
531	400	10	20	**132.77**	**148.81**	**12.08**

required for location 531 as opposed to 501. The final performance for location 531, measured on the train and validation set, is written in bold in the last row.

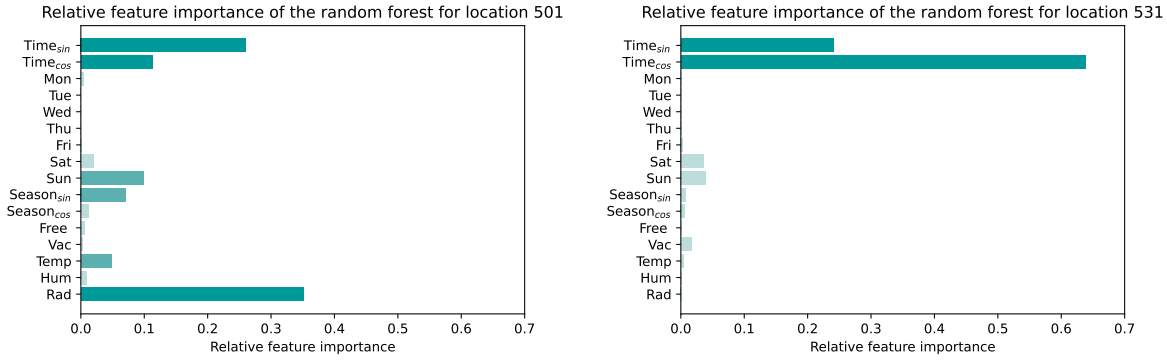
It should be noted that most likely an even better prediction model can be found by implementing a more extensive optimization. However, tuning the models is preferred, because it is computationally less expensive. In addition, the final parameters would have never been found by the optimization algorithm, because the performance on the validation set is also decreased. Because the performance of the final models is comparable on the train and validation set, the models obtained are assumed to be sufficient for this research. It is chosen not to further decrease δ_{rf}, because, in parallel, the performance on the validation set would degenerate further.

Relative feature importance

Inherent to the random forest is that the influence of each feature can be highlighted by the relative feature importance. Figure 5-1a and 5-1b show the feature importance of the final random forest model for locations 501 and 531, respectively. A decision tree is implemented for both locations, with the same hyperparameter, because it is comparable to and more intuitive than the random forest. The top splits of both trees can be found in Appendix A-3, which support the subsequent evaluations.

Interesting is the radiation, on which the first split for location 501 is made, and was not expected to have such high relative feature importance. However, the cluster analyses in Section 4-1, indicated that for this specific location a significant difference in traffic behavior was noticed during the summer and winter. On the other hand, location 531 was shown to be subject to a similar traffic behavior throughout the year, because it is mainly based on commuter traffic. This explains why the radiation and season are important for the first and not for the second location. This also highlights why the temperature is an important feature for location 501. That the radiation would be relatively more important than the temperature is in line with the correlations found in Section 4-2.

The time features are important for both locations. However, which one is most relevant differs. This can be explained using the sine and cosine functions, and the previous cluster analyses. The cluster centers indicated that for location 501, the main variance in traffic flow is caused by the afternoon peak that lies at the end of the day. This can be separated from the rest of the data set, by the sine function. On the other hand, for location 531, the morning and afternoon rush hours are significant. Therefore, first, the hours with a significant traffic



(a) Location 501 **(b)** Location 531

Figure 5-1: Relative feature importance of the random forest for location 501 and 531, in (a) and (b), respectively.

Table 5-3: Parameter space for Bayesian hyperparameter optimization of the multilayer perceptron.

Neurons	Layers	Learning rate	Batch size
$1, 2, \ldots, 200$	$1, 2, \ldots, 10$	0.00001, 0.0001, 0.001, 0.01	16, 32, 64

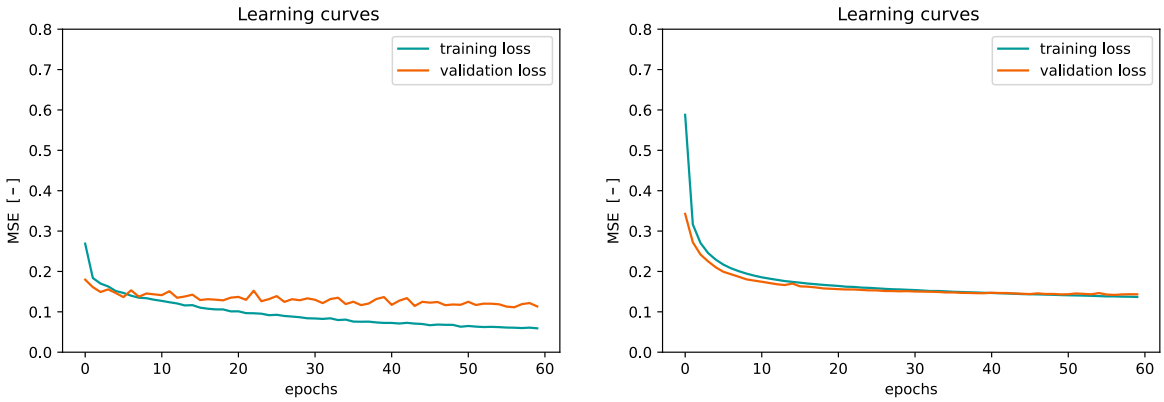
flow are grouped by the cosine function. Subsequently, the two peaks are separated by the sine function. The found decision trees indeed show this behavior.

At last, the dow features with a relatively high feature importance, correspond to the days that were shown to be subject to unique behavior. In addition, the vacation feature is important for location 531. Therefore, the behavior of the random forest complies with the insights obtained by the correlation analyses in the previous chapter.

5-2-2 Multilayer perceptron

The MLP is implemented with the `Keras.Model` library, using the Keras dense and concatenate layer. Bayesian parameter optimization is performed on the parameter space shown in Table 5-3, and the optimal parameters found after 100 evaluations for location 501 are given in the first row of Table 5-4. The model is trained on these hyperparameters, and the learning and validation loss curves are shown in Figure 5-2a. The difference between the training and validation loss curves shows that the model is overfitted on the training data. This is also indicated by the corresponding performance measures. Therefore, the model has to be tuned to counter this effect. This can be achieved by simplifying the model, by reducing the number of neurons and hidden layers. In addition, an appropriate learning rate should be found associated with the new hyperparameters. After a few iterations, it is chosen to implement the hyperparameters given in the second row of Table 5-4 and the corresponding learning curves are shown in Figure 5-2b.

Again, a similar optimization and evaluation are performed for location 531. The parameters found by Bayesian parameter optimization and the final hyperparameters, with corresponding performance measures, are given in the third en fourth row of Table 5-4, respectively. Because



(a) Optimized parameters **(b)** Final parameters

Figure 5-2: Learning curves during training of the multilayer perceptron for location 501, based on the initial optimized parameters and the final hyperparameters, in (a) and (b), respectively.

Table 5-4: Hyperparameters and performance measures of the multilayer perceptron for locations 501 and 531.

Location	Neurons	Layers	Learning rate	Batch size	$\text{RMSE}_{\text{mlp,train}}$ $[\frac{\text{veh}}{\text{h}}]$	$\text{RMSE}_{\text{mlp,val}}$ $[\frac{\text{veh}}{\text{h}}]$	δ_{mlp} [%]
501	63	3	0.001	16	49.78	75.02	50.71
501	55	2	0.0001	16	**81.96**	**84.37**	**2.95**
531	78	4	0.001	16	75.34	136.42	81.09
531	70	2	0.0001	16	**125.80**	**140.27**	**11.50**

the performance on the train and validation set is comparable, these models are assumed to be sufficient for this research and will be further evaluated in Chapter 7.

5-3 Summary

This chapter focused on converting the relevant information found in Chapter 4 into the final feature set and implementing the baseline prediction models.

The input information is divided into temporal periodic, temporal categorical, and weather features. Deep learning models often assume that values close to each other have a higher correlation than values further apart. Therefore, this assumption should only be imposed when valid. In addition, a generic input feature set is desired, to easily extend to other locations. Consequently, the time and season are each represented by two continuous cyclical features to capture the periodicity. In addition, the dow is represented by 7 categorical features, because which days are similar is location-dependent and not necessarily consecutive. Moreover, whether a day is a national holiday or during vacation is represented by two categorical features as well. At last, the weather features are implemented as three standardized continuous features, such that the final input is in $\in \mathbb{R}^{16}$.

Next, the total data set is divided into a train, validation, and test set, based on three

requirements. First, the validation and test set should be a representation of the entire data set. Secondly, data in the test set should not be indirectly implemented during training, through the look back or multistep predictions in the transformer. At last, the same division is desired for both baseline models and the transformer. Therefore, the first two years are taken as the training set, of which a random 20% is allocated to the validation set. Moreover, the last year is taken as the test set.

Two baseline models are implemented that do not take auto-correlation into account, the random forest and MLP. Bayesian hyperparameter optimization is implemented for both models and locations. The number of evaluations is constrained at 100 because it is a computationally expensive evaluation. Moreover, each model is examined on the validation set.

The investigated hyperparameters for the random forest are the number of estimators, minimum samples per leaf, and maximum depth. However, implementing the found hyperparameters indicated that the model was overfitting for both locations, due to the significant decrease in performance when evaluating on the validation set instead of the training set. Therefore, the hyperparameters are tuned to simplify the prediction model to decrease the chance of overfitting. This has been achieved by increasing the number of estimators and the minimum samples per leaf, and decreasing the maximum depth.

Next, the relative feature importance of the random forests is investigated. To make these more intuitive, a decision tree is implemented for both locations. The relative feature importance's, in addition to the first splits made by the decision tree, provided an intuitive explanation of the important features that corresponds to the insights found in the correlation analyses of the previous chapter. First, the radiation and season are important for location 501, because days during summer and winter are subject to different traffic behavior, whereas for location 531 they are not. Moreover, both time features are important, but which one is more important is reversed. This has been explained by the cluster centers found for both locations that showed that similar hours were found at different hours of the day for both locations. In addition, important dow features correspond to days that were found to have different behavior in the clustering analyses. At last, as expected, the vacation is found to be important for location 531.

Bayesian hyperparameter optimization is also implemented for the MLP on the number of hidden layers, the number of neurons per layer, the batch size, and learning rate. Again, overfitting occurred, indicated by the decrease in performance. Therefore, the hyperparameters are tuned by decreasing the number of layers, neurons, and learning rate.

It should be noted that most likely the baseline models can be improved by further optimizations. However, because the optimization is computationally very expensive, tuning the prediction model in the found range of hyperparameters is preferred. In addition, after tuning the hyperparameters, the performance on the validation set is also slightly decreased. Therefore, these hyperparameters would never be distinguished by the optimization algorithm. Because the performance of the final models is comparable on the train and validation set, these models are assumed to be sufficient for this research and will be evaluated and compared to each other and the transformer in Chapter 7.

Chapter 6

Transformers for long-term traffic flow predictions

This chapter focuses on the transformer. First, the implementation of the transformer is briefly discussed. Next, a baseline transformer, purely based on the traffic flow and time of the day, is optimized and implemented. Next, Section 6-3 extends this model step by step, by including additional external features, found to be important in Chapter 4, to investigate the influence of these features. Section 6-4 performs a new optimization for the transformer including all features and the final transformer is obtained. In addition, dropout is implemented as a regularization technique, to counter the effect of overfitting. Next, Section 6-5 makes the transformer behavior more intuitive, (1) to indicate the necessity of the model components implemented and (2) to increase the understandability of the results, which are discussed and compared with the baseline models in Chapter 7.

6-1 Transformer implementation

The transformer is implemented in python, with the publicly available `Keras library` [10]. The Keras.Model class is used to define the transformer, such that multiple layers can be grouped into an object with training and inference features. The multiple components, which form the transformer, are modeled as separate classes, defined as Keras.layer.Layers objects. It is chosen to program in an object-oriented way, such that extensions can easily be added, and it is simple to make adjustments to the model, due to the modularity. In addition, two standard layers from the Keras library are used, the dense and normalization layer. The algorithm shown in Appendix A-4 illustrates how the entire transformer model is built up from separate components. Moreover, the code used to build, train, and evaluate the models is available at `https://github.com/carmenpetsch/Transformer.git`.

Table 6-1: Parameter space for Bayesian hyperparameter optimization of the baseline transformer.

n_{heads}	Layers	d_{ff}	Learning rate	Batch size
$1, 2, 4$	$1, 2, \ldots, 10$	$10, 20, \ldots, 400$	$0.00001, 0.0001, 0.001, 0.01$	$16, 32, 64$

6-2 Transformer based on auto-correlation and time features

A baseline transformer is set up, which only takes the traffic flow and the time of the day into account. In addition, one feature is added, which inserts information about the relative position of the input. As an illustration, an encoder input of time t is constructed as $x_{\text{enc},t} = \left[\text{y}_t \ \text{time}_{\text{sin},t} \ \text{time}_{\text{cos},t} \ \text{age}_l\right]$, where l denotes the lookback. Moreover, based on the auto-correlation analysis of traffic flow in Section 4-3, the encoder input sequence length is set to 48, such that each input sequence has a dimension of $\mathbb{R}^{48\times 4}$. It is chosen to start with a simple transformer and extend these with extra features, to clearly investigate the influence of each additional feature.

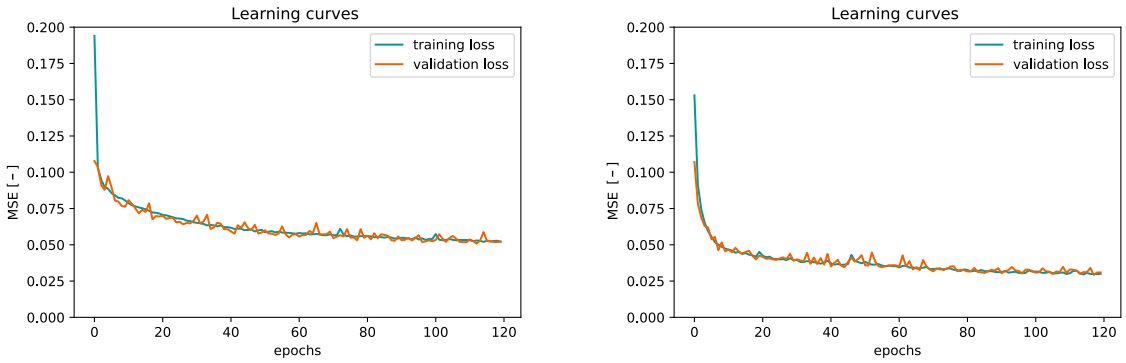
6-2-1 Set up of baseline transformer

Bayesian hyperparameter optimization is implemented for the baseline transformer for location 501, on the parameter space shown in Table 6-1. The parameter space is based on hyperparameters often used in state-of-the-art literature. The modulus of the dimension and number of heads (n_{heads}) should be zero. Therefore, the number of heads cannot equal three. After 100 evaluations, the best parameters found are: 2 heads, 6 layers, a d_{ff} equal to 200, a learning rate of 0.001, and a batch size of 16. The parameter space investigated is assumed to be sufficient because the found parameters do not lie on the boundary of the parameter space. Implementing these hyperparameters gives a total of 23.333 trainable parameters, calculated with (2-12), of which 11400 in the encoder, 11928 in the decoder, and 5 in the output layer.

It is chosen not to perform a separate optimization for location 531 but to implement the same hyperparameters, due to the large computational effort. The transformers are trained, and the training and validation loss curves are shown in Figure 6-1a and 6-1b for locations 501 and 531, respectively. A first indication is given that the models are not overfitting, because the validation and training loss curves are aligned. If desired, the small fluctuations in the validation loss curve can be reduced by decreasing the learning rate. However, this will increase the required number of computations to acquire a similar performance.

6-2-2 Preliminary results of the baseline transformer

The root mean squared error (RMSE) of the baseline models on the validation set, is indicated by the dark blue lines in Figure 6-2a and 6-2b, for locations 501 and 531, respectively. The models are optimized to predict one hour ahead. During training, the performance on all horizons approximately equals the first performance measure, which is the lowest. However, during inference, previous predictions are inserted to predict further horizons. As a result,



(a) Location 501 **(b)** Location 531

Figure 6-1: Learning curves during training of the baseline transformers, based on the optimized hyperparameters, for locations 501 and 531, in (a) and (b), respectively. The blue and orange graph indicate the training and validation loss curve, respectively.

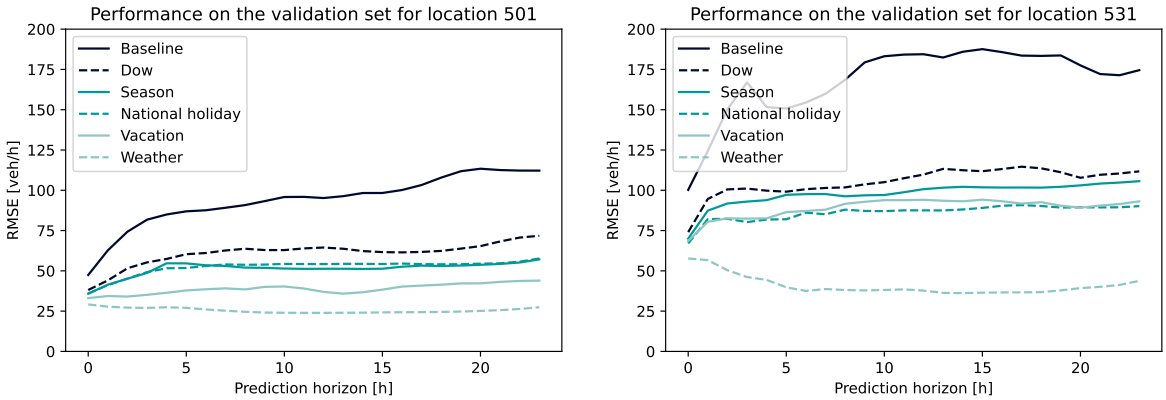
error accumulation occurs, which explains the decrease in performance over an increase in the horizon.

The error propagation does not seem smooth and has a few unexpected ups and downs. On the other hand, when investigating the error propagation on the training set, smooth curves were found with similar values. The traffic flow depends on the hour of the day, and will in general be higher during the day than during the night. The performance is measured in $\frac{veh}{h}$. Consequently, this will be higher during the day in general as well. Because a random 20% is taken as the validation set, the predictions corresponding to a certain prediction hour might include relatively few predictions of significant traffic flow, implying a lower RMSE. On the training set, this is less likely, which explains why this curve is smoother, and therefore, there should not be emphasized too much in the small ups and downs.

However, interesting is the increase in performance around a prediction horizon of 6 hours for both locations. This seems counter-intuitive at first, but this originates from the available self-attention in the decoder and is elaborated on in Section 6-5. The figures indicate a relatively large error for location 531. However, in Chapter 3, location 531 was shown to be subject to significantly more traffic flow than location 501. Moreover, the evolution of the errors over the prediction horizon seems to follow a similar pattern. Therefore, the behavior of the models for different locations seems comparable. The difference between the performance measures in the learning curves and the characteristics is because the performance on multiple prediction horizons is based on the actual traffic flow, whereas the model is trained on standardized traffic flow values.

6-3 Incorporation of external features in the transformer

This section investigates the influence of extending the baseline transformers, designed in the section above, with features identified in Chapter 4. First, the model is extended by including temporal periodic features, containing the day of the week (dow) and the season. Next, temporal categorical features, such as school vacations and whether a day is a national



(a) Location 501 **(b)** Location 531

Figure 6-2: Root mean squared error of the baseline transformer, and transformers with external influences incorporated, on different prediction horizons. The models for locations 501 and 531 are evaluated on the validation set in (a) and (b), respectively.

holiday, are included. Finally, the model is extended with weather features.

The hyperparameters found in the previous section are used. In reality, the performance can be improved by performing new optimizations, for each transformer. However, due to computational constraints, the previously found hyperparameters are assumed to be suited for the following extensions as well. As an exception, the number of heads is occasionally decreased to one, when the total number of features is odd. In addition, the number of epochs is limited to 60 for these evaluations, due to computational constraints as well.

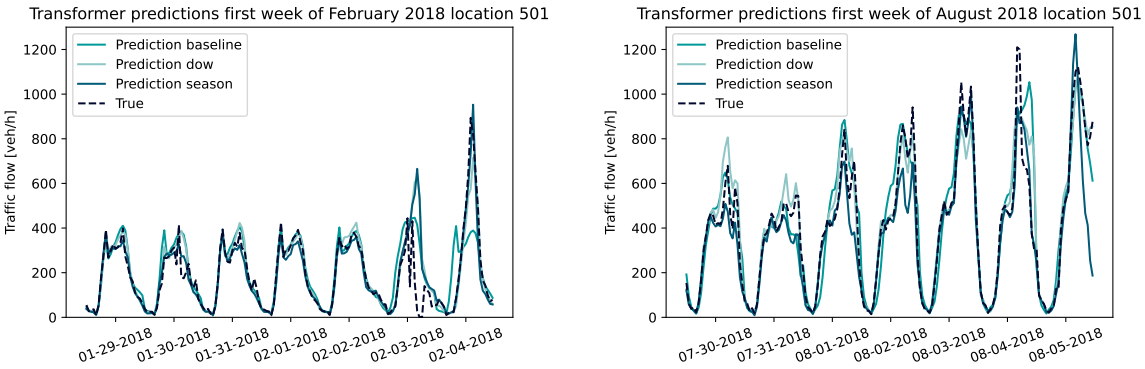
The results are not based on the test set and are meant to indicate that the transformer can work with and benefit from the external features. However, no investigation has been done yet regarding the overfitting of the models. Therefore, there should not be emphasized too much on the exact performance.

6-3-1 Temporal periodic features

For genericity of the model, the dow is included as seven categorical features, each representing a specific dow. As a result, the dimension of the input feature is increased from 4 to 11. The season of the year is included as two continuous features. By representing the season in this way, additional information, such as the month of the year, is indirectly included as well.

Figure 6-2 indicates that the performance increases significantly compared to the baseline transformer by including the dow for both locations. On the other hand, the model performance is only slightly increased for location 531, and even a bit decreased for location 501 by additionally including the season.

The prediction made by all three transformer models for location 501, for the first week of February and the first week of August in 2018, are shown in Figure 6-3. Each day, at midnight, a prediction is made for the next 24 hours. The importance of the dow is shown in the left figure, which illustrates that the baseline transformer is unable to predict the traffic behavior on Sunday, whereas the other transformers are able to predict the relatively high traffic flow.



(a) First week of February 2018 **(b)** First week of August 2018

Figure 6-3: Traffic flow prediction of the baseline transformer and transformers with temporal periodic features. A prediction for the next 24 hours is made at midnight each day, for the first week of February and August 2018 for location 501, in (a) and (b), respectively.

There will be briefly elaborated on the difference in including these temporal periodic features. Recall that overall the weekdays are similar and the weekend differs. Based on the previous traffic flow measurements, without the dow feature, the transformer is unable to know whether the predicted day is during the week or on the weekend. On the other hand, all days in summer experience a different behavior than in winter. For example, Chapter 4-1, indicated that the traffic flow seems to increase during summer for location 501. Figure 6-3 illustrates that the transformers, without the season feature, are also able to predict a higher traffic flow in summer than in winter. Therefore, without the season feature, the transformer is already able to incorporate the seasonal information. It is reasoned, that this information is indirectly incorporated in the traffic flow of the previous days. However, because implementing the feature is not detrimental for the performance, it is chosen to be incorporated for the genericity and comparability with the baseline models of Chapter 5.

6-3-2 Temporal categorical features

Different traffic flow behavior is noticed during national holidays, such as the first of January. By including this feature, the model can anticipate this irregular behavior. Figure 6-4 shows the predictions for the first of January. This highlights the positive effect of including the national holiday as a feature. In addition, Figure 6-2 shows that the performance indeed increases when the national holiday is added as a feature. Moreover, school vacations are also found to be beneficial for location 501. Although a different traffic behavior was noticed during the vacation, this feature seems less effective for location 531. This may have the same explanation as given for the season feature.

6-3-3 Weather features

At last, the temperature, radiation, and relative humidity are included in the transformer. Figure 6-2 shows an even further increase in performance by including the weather features. Interesting to see is that the predictions become even better at further horizons. This is

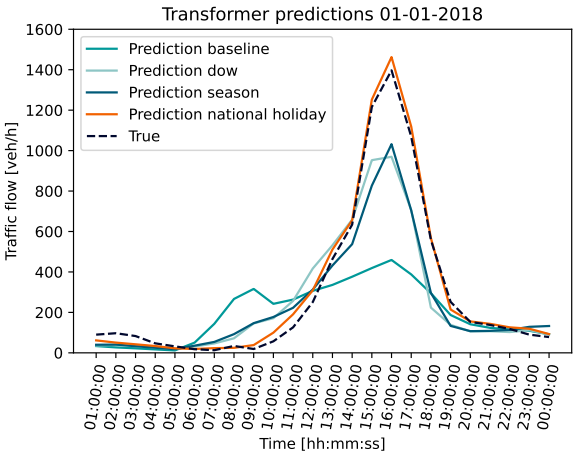


Figure 6-4: Traffic flow prediction for the first of January, made at midnight. The orange graph show the predictions made by the transformer, additionally including the national holiday.

counter-intuitive because previous error-prone predictions are inserted during multistep predictions. However, the advantage of further horizons is that more information is inserted into the prediction model through the decoder input. This implies that in this case, having more knowledge about the decoder input features has a greater effect than the error accumulation. This behavior is shown in Section 6-5, which focuses on gaining insights into the transformer behavior.

6-4 Final transformer models

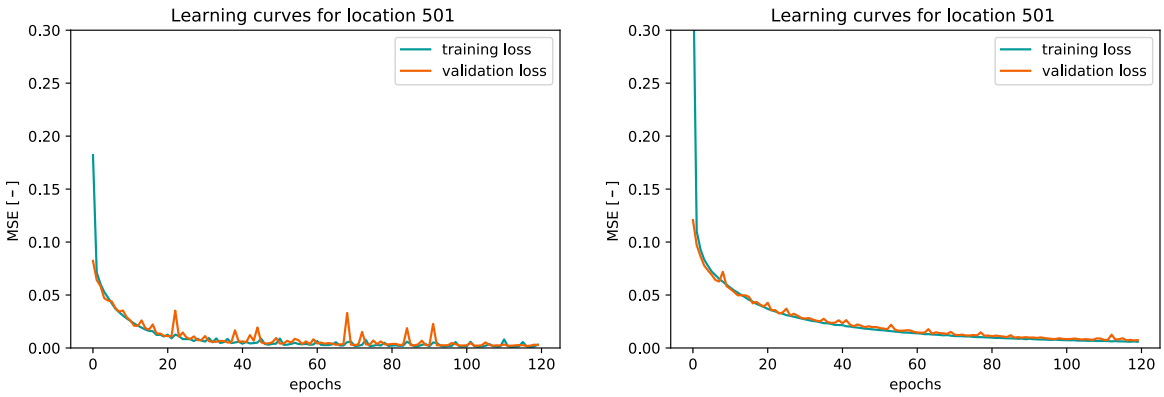
This section focuses on the transformer, including all external features. First, Bayesian parameter optimization is performed for both locations, including all external features. Moreover, the input sequence length is kept at 48, such that each encoder input sequence has a dimension of $\mathbb{R}^{48\times 18}$. Note that the extra two dimensions compared to the feature set for the baseline prediction models come from the additional traffic flow and age feature. Next, the preliminary results and insights into the behavior of the transformer model are obtained. The final transformer models are all trained on a maximum of 120 epoch, due to computational constraints. However, because convergence is not shown yet, the models can be improved a bit further. Because an indication of overfitting is found while evaluating the transformers, the influence of implementing dropout as a regularization technique is discussed.

6-4-1 Set up of the final transformer

Bayesian parameter optimization is implemented for locations 501 and 531, for the transformers, based on the entire feature set. The investigated parameter space equals the parameter space in Table 6-1, except for the possible number of heads (n_{heads}) which is set to $n_{\text{heads}} \in \{1, 2, 3, 6, 9, 18\}$. The found model hyperparameters are given in the first and second row of Table 6-2. Interesting to see is that these are almost equivalent, which indicates that a similar transformer can be implemented for locations subject to different traffic behavior.

Table 6-2: Optimized hyperparameter for the final transformers.

Location	n_{heads}	Layers	d_{ff}	Learning rate	Batch size
501	3	5	360	0.001	16
531	3	4	340	0.001	16



(a) $l_{\text{r}} = 0.001$ **(b)** $l_{\text{r}} = 0.0001$

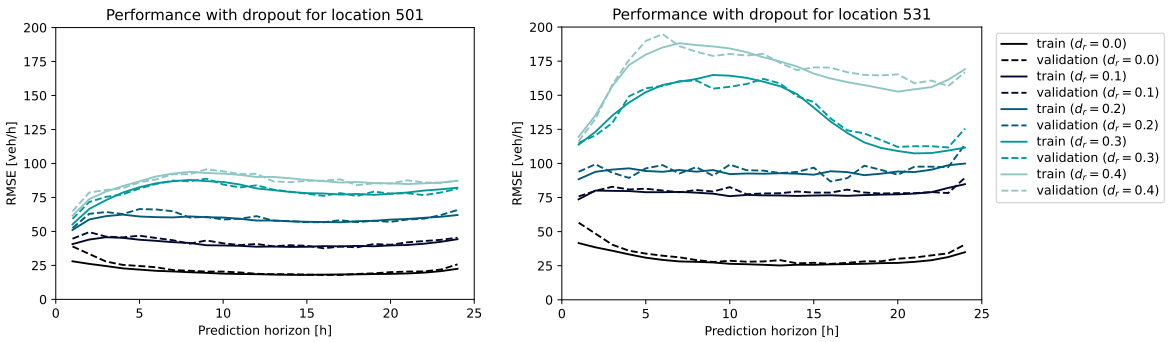
Figure 6-5: Learning curves during training of the transformer for location 501, constructed by the optimized hyperparameters and a learning rate (l_{r}) of 0.001 and 0.0001 in (a) and (b), respectively.

Implementing these hyperparameters gives a transformer model with 154.819 and 117,939 trainable variables for locations 501 and 531, respectively. The transformers are trained on these hyperparameters, and the training and validation loss curves for location 501 are shown in Figure 6-5a. The volatile validation loss indicates that the optimization steps might be too big. Similar curves were found for location 531. Therefore, it is chosen to decrease the learning rate to 0.0001, the corresponding learning curves are shown in Figure 6-5b. The performance obtained after the same number of epochs is slightly worse. However, a smoother learning curve provides more confidence in the performance of the model on new data, which is preferred. In addition, the training and validation loss curves are aligned, which again indicates that the model is not overfitting the training data.

6-4-2 Preliminary results of the final transformer

The performance of the transformers is concisely investigated in this section. The RMSE for each prediction horizon is calculated for the train and validation data for both locations. The decrease in performance is calculated by the relative increase in the error, represented by $\delta_{\text{v},1}$ and $\delta_{\text{v},24}$, which represent the increase at the first and last prediction step between the training and validation set and are given in Table 6-3, corresponding to a dropout rate (d_{r}) equal to zero, on which there will be elaborated on shortly.

These values illustrate that the performance is decreased when the model is applied to the validation. This is expected because the transformer has not seen this data before. However, the increase is significant and indicates that the transformer model is overfitting the training



(a) Location 501 **(b)** Location 531

Figure 6-6: Root mean squared error of the transformers for both locations, with a dropout rate (d_r) equal to 0.0, 0.1, 0.2, 0.3, and 0.4 on the training and validation set, represented by the solid and dotted line, respectively.

data. In Chapter 5, the effect of overfitting was decreased by simplifying the baseline models. However, the transformer is subject to more hyperparameters, making it time-intensive to tune. Due to these difficulties, it is chosen to investigate regularization techniques instead, to reduce the chance of overfitting.

6-4-3 Extension of the final transformer with dropout

One of the most commonly used regularization techniques in transformers is applying dropout [21, 58], first proposed in [54]. A dropout layer randomly sets the output of certain neurons to zero. The chance of a neuron being dropped is defined by the dropout rate (d_r). By implementing dropout, the prediction model becomes less sensitive to specific neurons, generalizes better, and is less likely to overfit. A dropout layer is implemented after each multi-head attention, feedforward, and positional encoding layer. The dropout rate is a hyperparameter. However, when another hyperparameter optimization is performed, it is found that the algorithm sets the dropout rate to zero because again the validation loss does not indicate overfitting, and applying dropout is found to degrade the performance on the validation set. Note that this might be because the models are not fully converged yet and the transformer with dropout requires more steps to retrieve the same performance.

Therefore, there is looked into commonly used dropout rates in the literature, which illustrates that a dropout rate of 0.1, 0.2, 0.3, and 0.4 are commonly used in the transformer [21, 58]. These dropout rates are implemented and the effect on the train and validation set is investigated. It is chosen not to look into the effect on the test set because this would imply that the transformer is tuned on the test set. In Figure 6-6, the corresponding RMSE on different prediction horizons are shown for locations 501 and 531. The next section compares the different models with and without dropout.

6-4-4 Comparison of transformer with and without dropout

A difference in performance is found between the models with dropout and the original transformer without dropout, as shown in Figure 6-6. The graphs indicate, that the transformer

Table 6-3: Performance measures transformer with and without dropout.

	Dropout rate (d_r)	$\delta_{v,1}[\%]$	$\delta_{v,24}[\%]$	$RMSE_{v,1}[\frac{veh}{h}]$	$RMSE_{v,24}[\frac{veh}{h}]$
Location 501	0.0	39.00	14.46	38.94	25.74
	0.1	9.73	2.51	44.61	45.39
	0.2	**2.71**	**5.98**	**52.45**	**65.76**
	0.3	7.25	-0.91	59.20	81.43
	0.4	5.20	0.31	64.78	87.49
Location 531	0.0	35.98	16.21	56.56	40.57
	0.1	**2.91**	**5.59**	**75.82**	**89.53**
	0.2	6.19	14.02	93.89	113.91
	0.3	1.11	12.53	115.08	125.54
	0.4	-2.65	-1.32	116.36	166.83

performs better without dropout layers on the training and validation set. However, to reduce $\delta_{v,1}$ and $\delta_{v,24}$, dropout layers might be beneficial. These characteristics are provided in Table 6-3 for the different models, in combination with $RMSE_{v,1}$ and $RMSE_{v,24}$.

The performance measures indicate that $\delta_{v,1}$ and $\delta_{v,24}$ are reduced significantly by implementing dropout. This implies that the transformers with dropout are less overfitted on the training data. For location 501, a d_r of 0.2 holds the lowest $\delta_{v,1}$ value. Therefore, this model is thought to be most suited. On the other hand, a higher dropout rate results in a lower $\delta_{v,24}$ value. However, because the model is trained on a prediction horizon of one hour, the latter performance characteristic is assumed to be less crucial.

Table 6-3 indicates the largest decrease in $\delta_{v,1}$ and $\delta_{v,24}$ for location 531, by implementing a d_r of 0.3 and 0.4. However, Figure 6-6b implies that this comes at the cost of the RMSE. Therefore, for location 531, a d_r of 0.1 is thought to hold the best performance, because a decrease in $\delta_{v,1}$ and $\delta_{v,24}$ is induced without significantly decreasing the performance. Therefore, the transformers highlighted in bold in Table 6-3 are chosen as the final transformers.

To conclude, the final transformer model is based on the hyperparameters obtained by the Bayesian hyperparameter optimization. Subsequently, the learning rate is reduced to 0.0001 to obtain smoother learning curves. At last, dropout layers with d_r equal to 0.2 and 0.1 for locations 501 and 531, respectively are implemented. By implementing these layers, the relative decrease in performance by implementing the model on the validation set as opposed to the training set is decreased. The performance on the test set is evaluated and compared to the baseline models in the next chapter, but first insights into the behavior of the transformer will be provided.

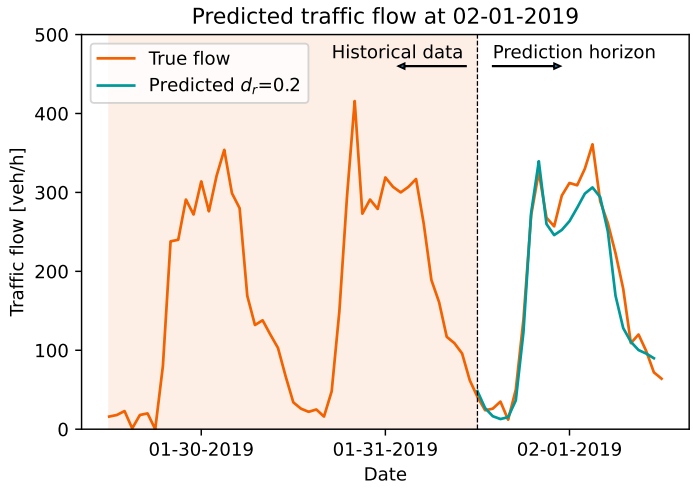


Figure 6-7: Prediction for location 501 made by the final transformer for 02-01-2019. The highlighted part indicates the available historical traffic flow. In addition, the orange and blue graphs illustrate the true and predicted traffic flow, respectively. Moreover, the dashed line indicates the time (t), when the predictions are made.

6-5 Insights into the transformer behavior

This section makes the behavior of the transformer, designed in the previous sections, more intuitive with the goal (1) to indicate the necessity of the different model components and (2) to increase the understandability of the results in the next chapter. Similar observations are made for both locations. Therefore, there is only elaborated on location 501 in the remainder of this section.

This is done in line with an example; the traffic flow prediction of 02-01-2019, starting at midnight, which lies in the test set and is based on the historical traffic flow of 01-30-2019 and 01-31-2019. The corresponding (historical) traffic flow and the predictions made by the transformer are shown in Figure 6-7 by the orange and blue graph, respectively. Moreover, the dashed line represents the start time (t) of the prediction.

In Section 2-4 the structure of the transformer was explained, which showed that attention is applied in three different ways. Self-attention is applied in the encoder and masked self-attention in the decoder. In addition, attention is applied between the encoder and decoder. The attention weights applied for this specific example are elaborated on in the next subsection. It should be noted that the attention weights depend on the specific inputs. Therefore, the weights shown are representative of the working principles of the transformer, but will not be identical in other implementations. In addition, the inputs are transformed multiple times. Therefore, after the first attention layer, the relative time indices do not correspond to the exact time but to one containing information of multiple time indices. However, for comprehensibility, the time relative to t is set on the axis of the figures.

(a) Layer 2 head 1 **(b)** Layer 2 head 2 **(c)** Layer 2 head 3

Figure 6-8: Self-attention in the encoder for layer 2 and head 1, 2, and 3 in (a), (b), and (c), respectively. Where a darker blue indicates that more attention is paid to the corresponding input. The y- and x-axis represent the previous inputs relatively to t, starting from 48 hours ago.

6-5-1 Self-attention in the encoder

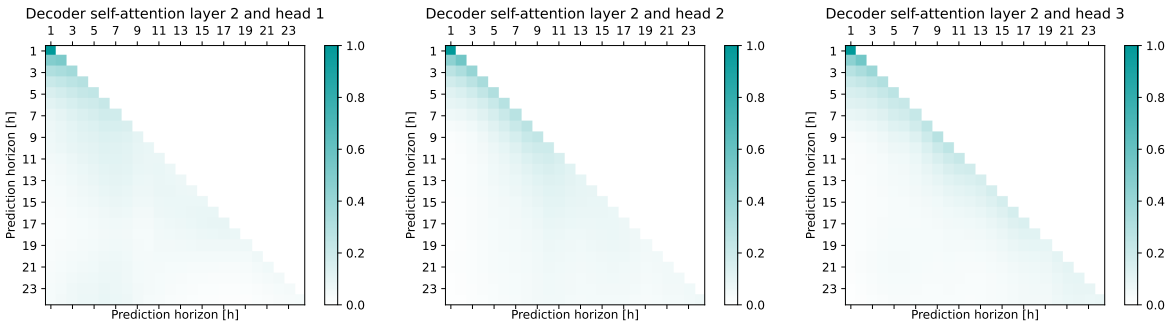
The self-attention in the encoder looks into the historical data. The final transformer is composed of 5 layers and 3 attention heads in the encoder and decoder. Therefore, a total of 15 attention weights are obtained in the encoder, each of dimension $\mathbb{R}^{48\times48}$. The first layer is a general layer, which attends to all other inputs. However, the other layers are formed by multiple blocks and vertical lines, as illustrated in Figure 6-8, which shows the attention weights of the multiple heads in layer two. The vertical lines indicate that all inputs strongly attend to the same inputs. This allows the head to focus on a specific behavior. On the other hand, blocks on the diagonal indicate that attention is paid to itself and other neighboring inputs. Interesting to see is that the transformer separates the two historical days. In addition, it is found that when the prediction is made a few hours later, similar blocks are found, but shifted.

The figures indicate that the multiple attention heads indeed pay attention to different parts of the input. Moreover, in Appendix A-5 the weights of the other layers are additionally shown. These illustrate that each layer also attends to different parts of the input. This highlights the effectiveness of the multiple layers and heads in the transformer.

6-5-2 Self-attention in the decoder

Self-attention is also applied in the decoder, which includes future input features. The maximum prediction horizon is 24 hours. Therefore, each attention weight is of dimension $\mathbb{R}^{24\times24}$. Similar to the self-attention in the encoder, the first layer pays attention to most inputs. However, the other layers follow different patterns. Figure 6-9 shows the decoder self-attention for the second layer and head 1, 2, and 3, in (a), (b), and (c), respectively. In addition, the self-attention weights of the other layers are provided in Appendix A-5. These again indicate that different layers and heads pay attention to different parts of the input.

The white upper triangle is caused by the look-ahead mask, which prevents future information flow. This also highlights that predictions on the first few horizons are subject to fewer data than predictions on further horizons. This explains the increase in performance found after the first few predictions in Section 6-2-2. At first, this was counter-intuitive, because the



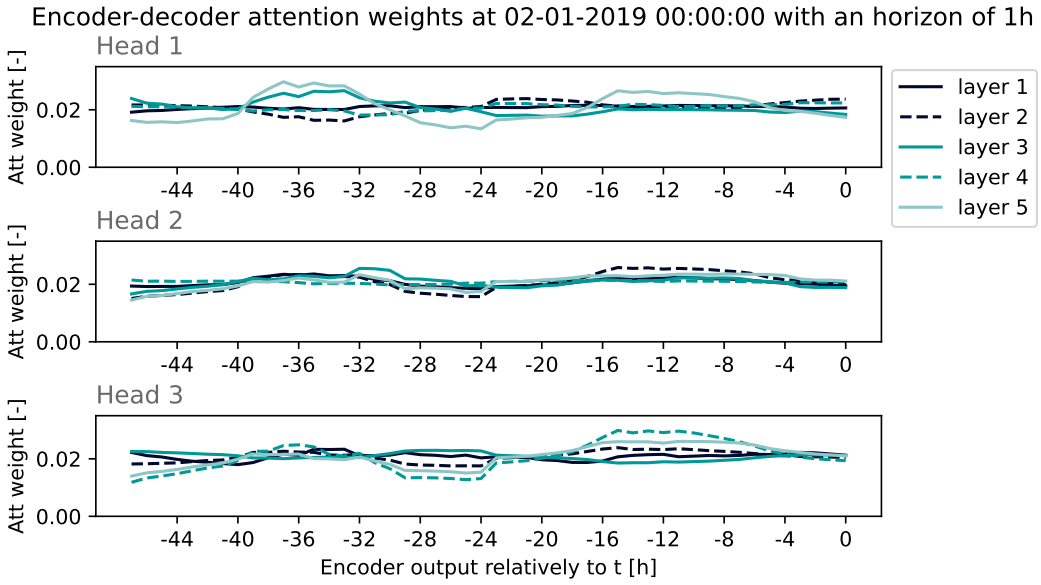
(a) Layer 2 head 1 **(b)** Layer 2 head 2 **(c)** Layer 2 head 3

Figure 6-9: Self-attention in the decoder for layer 2 and head 1, 2, and 3 in (a), (b), and (c), respectively. Where a darker blue indicates that more attention is paid to the corresponding input. The y- and x-axis represent the future inputs up to 24 hours. Moreover, the white upper triangle is caused by the look-ahead mask.

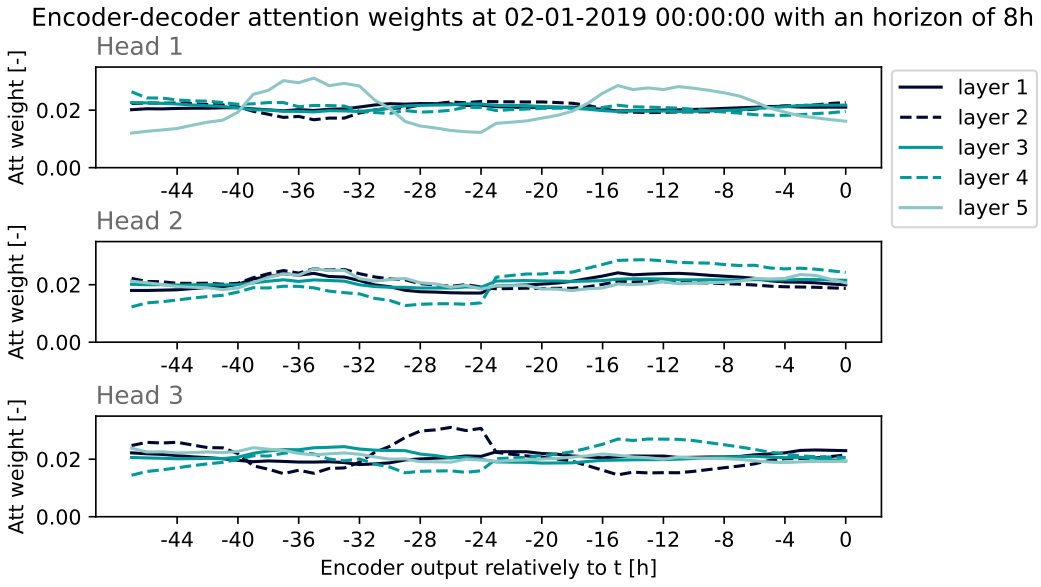
performance was expected to degrade due to error accumulation. However, this implies that the additional information in the decoder at further horizons is beneficial and even counters the negative effect of the error accumulation.

6-5-3 Attention between encoder and decoder

The last attention block is applied between the encoder output, set to the value and key, and for each layer the corresponding decoder self-attention output, which is set to the query. For comprehensibility, it is chosen not to show the entire encoder-decoder attention matrices. However, in Figure 6-10a and 6-10b, the encoder-decoder attention weights for the multiple heads and layers are shown for a prediction horizon of 1 and 8 hours, respectively. The three subplots represent the multiple heads. Additionally, the different layers are indicated by the blue graphs. These figures clearly show the difference between the multiple heads and layers. Moreover, the difference in the two figures indicates that indeed the attention weights depend on the inputs. It should be noted that the weights are not applied directly to the historical traffic flow corresponding to the same relative time, because multiple transformations are already applied in the encoder.



(a) Prediction horizon of 1 hour



(b) Prediction horizon of 8 hours

Figure 6-10: Encoder-decoder attention weights, with in (a) the prediction made for a horizon of 1 hour, and in (b) for a horizon of 8 hours. The subplots show attention heads 1, 2, and 3, respectively. Moreover, the multiple layers are indicated by the different blue graphs.

Table 6-4: Final parameters of the transformers.

Location	n_{heads}	Layers	d_{ff}	Learning rate	Batch size	Dropout rate
501	3	5	360	0.0001	16	0.2
531	3	4	340	0.0001	16	0.1

6-6 Summary

This chapter showed step-by-step how the final transformers for locations 501 and 531 have been obtained. First, a baseline transformer was implemented, purely based on the traffic flow, time, and age input feature. Next, the effect on the performance by including other external features was investigated, by extending the baseline transformer with one feature at a time. The dow was shown to increase the performance significantly, whereas the season feature had little effect. This difference is explained as follows, without the dow feature the transformer is unable to know the dow based on the previous information, whereas the effect of the season is shown in the historical data and is therefore indirectly already incorporated without the actual feature. The national holiday and vacation improve the performance on these days. However, in the total performance characteristics, the effect is negligible, because it concerns only a small part of the entire data set. At last, the weather features are shown to have a significant effect. It is chosen to incorporate all the features in all models, including the ones with little effect because the effect is shown not to be detrimental to the performance, and it is beneficial in terms of comparability en genericity of the models.

Next, Bayesian hyperparameter optimization is performed for both locations. After evaluations, it is chosen to decrease the learning rate to reduce the effect of the volatile validation loss. Moreover, the RMSE is investigated on the train and validation data for each prediction horizon. The transformers are found to overfit the train data. Therefore, dropout is implemented as a regularization technique.

Commonly used dropout rates in the literature are implemented and the performance on the train and validation set is compared. In the end, a dropout rate of 0.2 and 0.1 are chosen for locations 501 and 531, respectively, because these decrease the relative increase in error from the training to the validation set without significantly decreasing the performance on the validation data. The final transformer parameters are given in Table 6-4.

At last, an intuition behind the working principles of the transformer is obtained to indicate the necessity of the model components and to increase the understandability of the results in the next chapter. It was shown that the multiple layers and heads in the attention blocks indeed behave differently. In addition, based on the decoder self-attention weights, the evolution of the performance over the prediction horizons has been elaborated on.

Chapter 7

Results and comparison of baseline prediction models and transformer

The baseline prediction models and transformers have been designed in Chapter 5 and 6, respectively. This chapter will evaluate and compare these models on four aspects. First, the performance on the train, validation, and test set is investigated in Section 7-1. Next, the performance at different times of the day and prediction horizons is examined in Section 7-2. Next, days of the year, where the prediction models tend to be less accurate, are investigated in Section 7-3. At last, an estimate of the uncertainty of the predictions is provided by analyzing the distribution of the relative errors in Section 7-4, and an example of the final predictions is provided in Section 7-5.

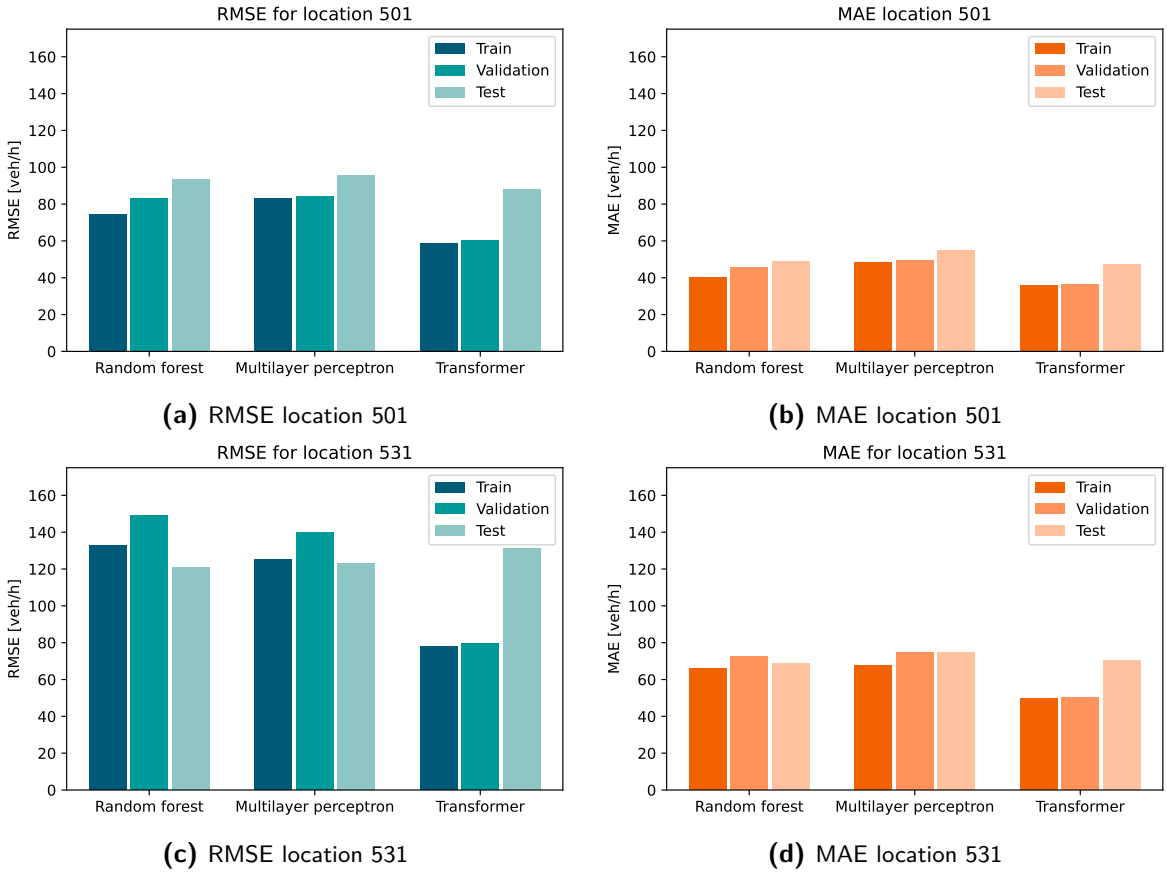
7-1 Performance on different data sets

The previous chapters focused on designing and implementing the prediction models. The performance of the prediction models was investigated on the train and validation set. Next, the model structures were tuned accordingly to prevent overfitting. This section focuses on the performance of the models on unseen data, the test set, relative to the performance on the train and validation set.

7-1-1 Comparison of the prediction models

In Figure 7-1 the root mean squared error (RMSE) and mean absolute error (MAE) of the random forest, multilayer perceptron (MLP), and transformer are shown for both locations.

The baseline prediction models have a similar performance on the test set, which indicates that there is no clear preference for either of the two based on these characteristics. However, for location 501 the random forest seems to overfit a bit more on the training set. This could be reduced by further tuning the hyperparameters as done in Chapter 5. However, as



(a) RMSE location 501

(b) MAE location 501

(c) RMSE location 531

(d) MAE location 531

Figure 7-1: Performance characteristics on the train, validation, and test set for the random forest, multilayer perceptron, and transformer for location 501 and 531.

stated before, the difference in performance on the train and validation set is thought to lie in acceptable ranges. Moreover, for location 531 both models seem to underfit on the data, because the models perform better on the test set. The highest errors are ascertained to occur during a specific period, which lies in the train and validation set. Section 7-3 examines the performance of the models throughout the year and further explores the implications that the test set contains relatively uncomplicated traffic behavior.

Figure 7-1a and 7-1b illustrate that the transformer outperforms the baseline prediction models for location 501 on the test set. On the contrary, Figure 7-1c and 7-1d, reveal that the transformer is outperformed by the baseline models for location 531. The next sections go into more detail regarding the prediction horizon, time of the day, and characteristics of the day that might cause either of the models to be preferred.

The difference in performance of the transformers on the train, validation, and test set is significant for both locations. This highlights that the transformer is overfitting on the training set, which is unexpected because the performance on the validation set did not indicate overfitting. This suggests that the validation set is not representative of the test set, which is discussed in the following.

7-1-2 Discussion overfitting of the prediction models

Figure 7-1 illustrates that the performance of the transformer decreases when the model is applied to the test set for both locations. This was anticipated because the models have not seen this data before. However, the performance on the test set is significantly worse than on the validation set, which is unexpected because the prediction models are neither based on the validation nor the test set. Therefore, a similar behavior was expected, which implies that the validation set is not a good representation of the test set. Accordingly two reasons are investigated. First, 2019 might not be a proper representation for the years 2017 and 2018, causing a discrepancy between the data sets. Secondly, the transformer might indirectly already be subject to the validation set through the historical traffic flow and multistep predictions. Recall, that the train and test set were divided as 2017/2018, and 2019, respectively, and a random 20% of the training set is taken as the validation set.

The first explanation is supported by the performance of the MLP and random forest for location 501 that indicates an inferior performance on the test set. This cannot be due to the second explanation because it does not apply to the baseline models. Therefore, to look into the discrepancy between years, the entire data set is shuffled and then divided. The performance of the corresponding baseline models, trained and evaluated on these data sets, is shown in Figure 7-2, by random forest 2 and MLP 2. The performance on the validation and test set are similar, which indicates that the difference in performance of the original baseline models could indeed be caused by a discrepancy between years.

Secondly, the possibility of implementing a different validation set in the transformer for location 501 is investigated. First, instead of randomly, the last 20% of the training set is taken as the validation set. The training and validation loss curves now indicate that the model is underfitting because the model performs better on the validation set. The correlation analyses in Section 4-1 indicated that for location 501, most irregular days occur in summer. The last 20% of the training set contains the winter period of 2018. Therefore, the validation set is easier to predict than the training set and no conclusions can be drawn. Secondly, every fifth week of 2017/2018 is taken as the validation set, such that the data, indirectly seen by the transformer through the historical traffic flow and multistep predictions, is limited. The same hyperparameters are implemented and the performance characteristics for the new transformer are shown in Figure 7-2 by Transformer 2. The transformer has a similar performance on the train set. However, the performance on the validation set now clearly indicates that the transformer is overfitting and is similar to the performance on the test set. This highlights that indeed randomly selecting the validation set is not representative for the test set and overfitting occurs, which explains the unexpected decrease in the original transformer performance on the test set.

Therefore, both explanations seem to influence the discrepancy between the performance on the different data sets. Shuffling the data is useful for the baseline prediction models. However, it is undesired to implement the shuffled data set in the transformer, because the test set will not be valid anymore. Moreover, for comparability, it is desired to apply the models on equivalent data sets. Therefore, it is chosen not to shuffle the data. Additionally, in reality, the discrepancy between the different years is less significant, because the model can be updated throughout the year, which will be discussed in the next chapter.

Applying the transformer to a different validation set is beneficial. Ideally, an entire year can be used as the validation set, such that all different types of traffic behavior are included,

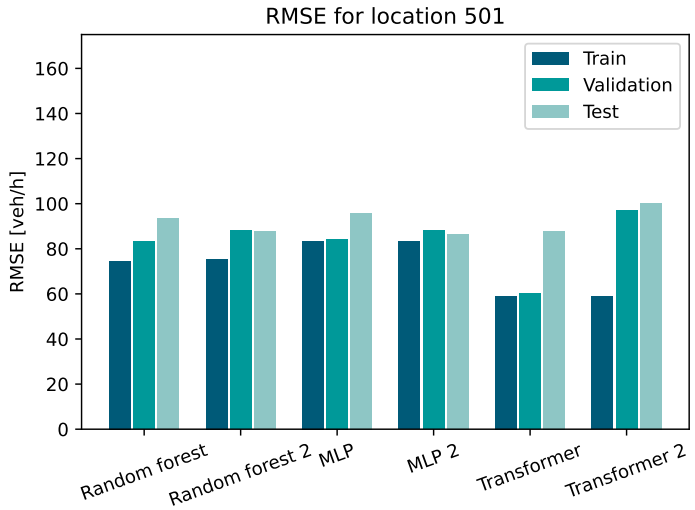
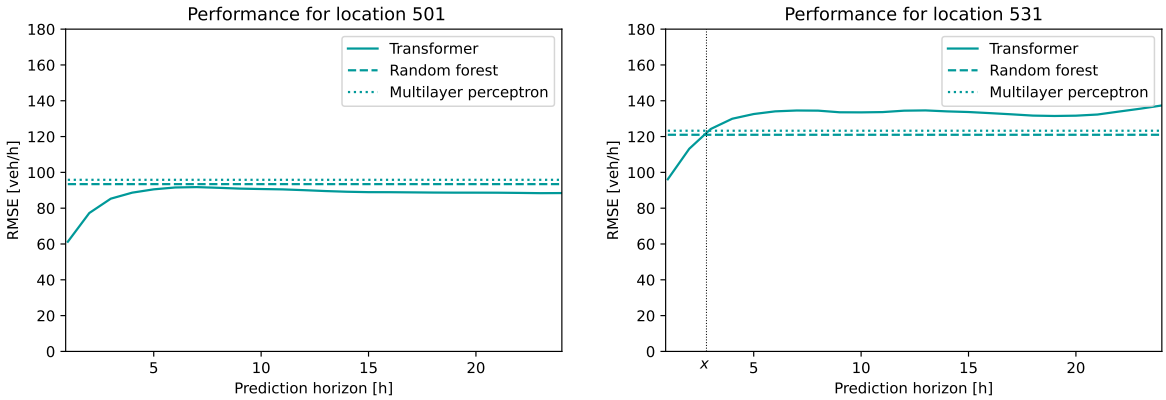


Figure 7-2: Comparison performance of the original and adjusted prediction model, where the latter are indicated by the 2 in the model name. For the adjusted baseline models shuffled data sets are implemented, which highlight the discrepancy in traffic behavior at different years. For the adjusted transformer, every fifth week of the training set is taken as the validation set, which shows that the original validation set is not representative of the test set.

without indirectly including the validation set during training. However, if the data is limited, a different validation set can be taken, such as described above. During hyperparameter optimization and training, the algorithm was not able to notice overfitting. Therefore, new hyperparameter optimizations should be performed based on the new validation set. In addition, tuning the model by aiming for a similar performance on the train and validation set would again reduce the chance of overfitting. Finding a more suitable validation set and designing a new transformer accordingly is expected to improve the model performance. For the scope of this research, a transformer, based on the new validation set is not further implemented. However, it is recommended to examine this possibility in future research.

7-2 Performance on different prediction horizons and time of day

One of the objectives of this research is to investigate whether the transformer is advantageous for long-term predictions. The previous section indicated that the baseline prediction models outperformed the transformer for location 531. However, this might not apply for all prediction horizons, because the performance of the transformer is thought to decrease over the prediction horizon due to error accumulation. Moreover, the influence of current events is also likely to decrease with the prediction horizon. In addition, in Chapter 4, traffic flow is shown to be highly dependent on the time of the day. Therefore, the behavior of the errors over the prediction horizon and additionally the time of the day is investigated. It is chosen not to focus on the relative error because this accentuates the errors made when the locations are subject to little traffic. For this research, these errors are less important because when the locations are subject to little traffic flow there will be less demand for traffic control.



(a) Location 501 **(b)** Location 531

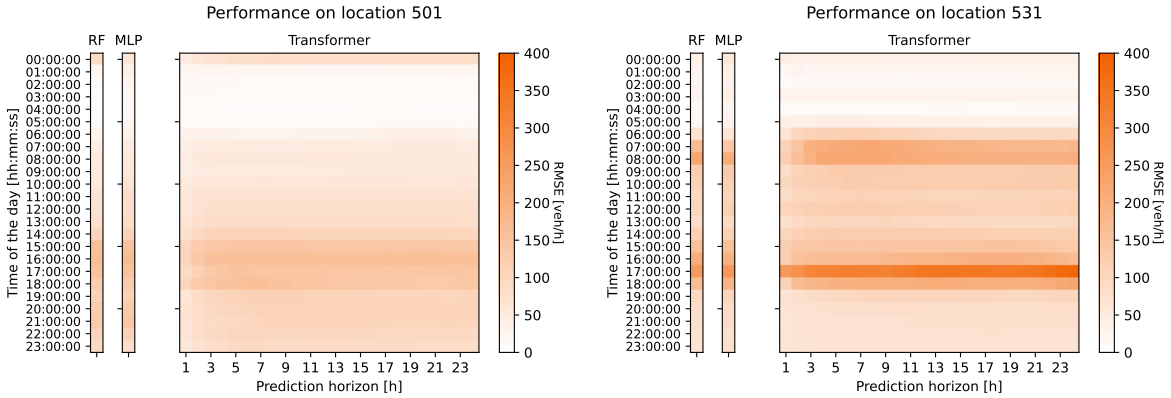
Figure 7-3: RMSE on the test set for the random forest, multilayer perceptron, and transformer for locations 501 and 531 in (a) and (b), respectively. Moreover, x represents the prediction horizon when the transformer is surpassed by the baseline prediction models.

7-2-1 Comparison of the prediction models

Figure 7-3b shows the evolution of the RMSE over the prediction horizon. The first figure indicates that for location 501, the transformer outperforms the baseline models on all prediction horizons but is especially superior in the first 6 prediction hours. For location 531, the transformer is advantageous up to a horizon of 3 hours, indicated by x in Figure 7-3b.

The traffic flow depends on the time of the day. Therefore, additionally, the RMSE corresponding to each time and prediction horizon on the test set is calculated for both locations and shown in Figure 7-4. A similar evolution of the errors was found for the MAE, which is therefore not shown. The figures illustrate that the different models encounter difficulties with similar times of the day. For location 501, the traffic flow around 16:00:00 seems most difficult to predict. For location 531, the highest errors occur around 08:00:00 and 17:00:00. This is related to the cluster centers found in Section 4-1, which show that the locations are subject to most traffic flow at these hours. Interesting is the significant error for location 531 at 17:00:00. In addition, the previous section showed that the baseline models had a better performance on the test set than on the train and validation set. By looking into the errors throughout the year, these are found to be related to each other. Therefore, there is elaborated on this in Section 7-3.

The figures indicate different types of behavior of the transformer at different times of the day. When subject to little traffic, the error of the prediction models is relatively low, similar, and constant over the prediction horizons. On the other hand, at times generally subject to relatively high errors, the error is shown to increases over the prediction horizon. Moreover, the transformer is especially superior in the first few prediction horizons. This difference in behavior at different times of the day is investigated in the following.



(a) Location 501 **(b)** Location 531

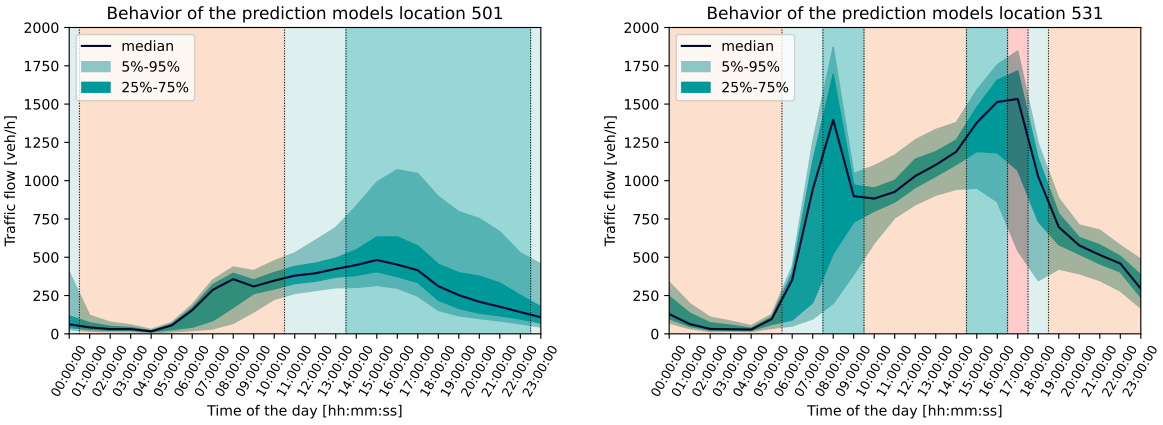
Figure 7-4: RMSE on the test set for the random forest, multilayer perceptron, and transformer. Shown for the different times of the day and prediction horizons for locations 501 and 531.

7-2-2 Discussion of the prediction models behavior

Figure 7-5 shows the median and 5%, 25%, 75%, and 95% quantiles of the traffic flow in the test set over the day for both locations. The colored rectangles highlight different types of behavior. These characteristics and behaviors are thought to be related to each other.

First, the orange rectangles highlight when the performance of the baseline models and transformer is similar. In addition, the performance of the transformer is equivalent over the prediction horizons. This occurs when the locations are subject to little traffic or regular traffic with a small deviation. For location 531 this behavior is often shown and also occurs during the day when there is quite a lot of traffic. This indicates that the transformer is not able to benefit from the additional information, even on a short horizon. On the other hand, for location 501 this behavior is only seen when subject to little traffic. Secondly, at the dark green rectangles, the transformer outperforms the baseline models at all prediction horizons, but is especially superior at short prediction horizons, similarly to Figure 7-3a. This behavior occurs at times of the day subject to a large range of traffic flows. At last, the hours of the day in between these extremes are highlighted by the light green rectangles. Here, the transformer outperforms the prediction models on the first few prediction horizons. On further horizons, the models have a similar performance, where interchangeably the one outperforms the other, similar to the pattern shown in Figure 7-3b. The red rectangle around 17:00:00 highlights when the transformer is outperformed by the baseline prediction models on all prediction horizons based on the RMSE but performs better based on the MAE. This behavior occurs once and is elaborated on in the next section.

Section 4-1 showed that location 531 is subject to commuter traffic and has a strong periodicity, whereas location 501 is subject to more irregular traffic behavior. This indicates that the transformer is advantageous on both short and long prediction horizons when subject to irregular traffic flow. However, when the traffic flow is caused by regular traffic flow, the advanced model is redundant and the transformer is only advantageous on shorter horizons. In addition, location 501 has many subsequent locations with a broader range of traffic flow values. Consequently, irregular behavior might be identified earlier, which highlights why for location 501 the transformer is superior for a relatively long prediction horizon in Figure 7-3.



(a) Location 501 **(b)** location 531

Figure 7-5: The characteristics of the traffic flow are shown by the median and 5%, 25%, 75%, and 95% percentiles. Different behavior of the models is highlighted by the colored rectangles. The orange, at 01:00:00-10:00:00 for location 501, and at 00:00:00-05:00:00, 10:00:00-14:00:00, and 19:00:00-23:00:00 for location 531, indicates that all prediction models have a similar performance that barely changes over the prediction horizon. At the light green, at 00:00:00, 11:00:00-13:00, and 23:00:00 for location 501, and 06:00:00-07:00:00 and 18:00:00 for location 531, the transformer outperforms the baseline models on short horizons and performs similar on further horizons. At the dark green at 14:00:00-22:00:00 for location 501 and 08:00:00-09:00:00 and 15:00:00-16:00:00 for location 531, the transformer is superior in all prediction horizons. At the red rectangle at 17:00:00 the transformer has a worse RMSE and better MAE.

7-3 Performance throughout the year

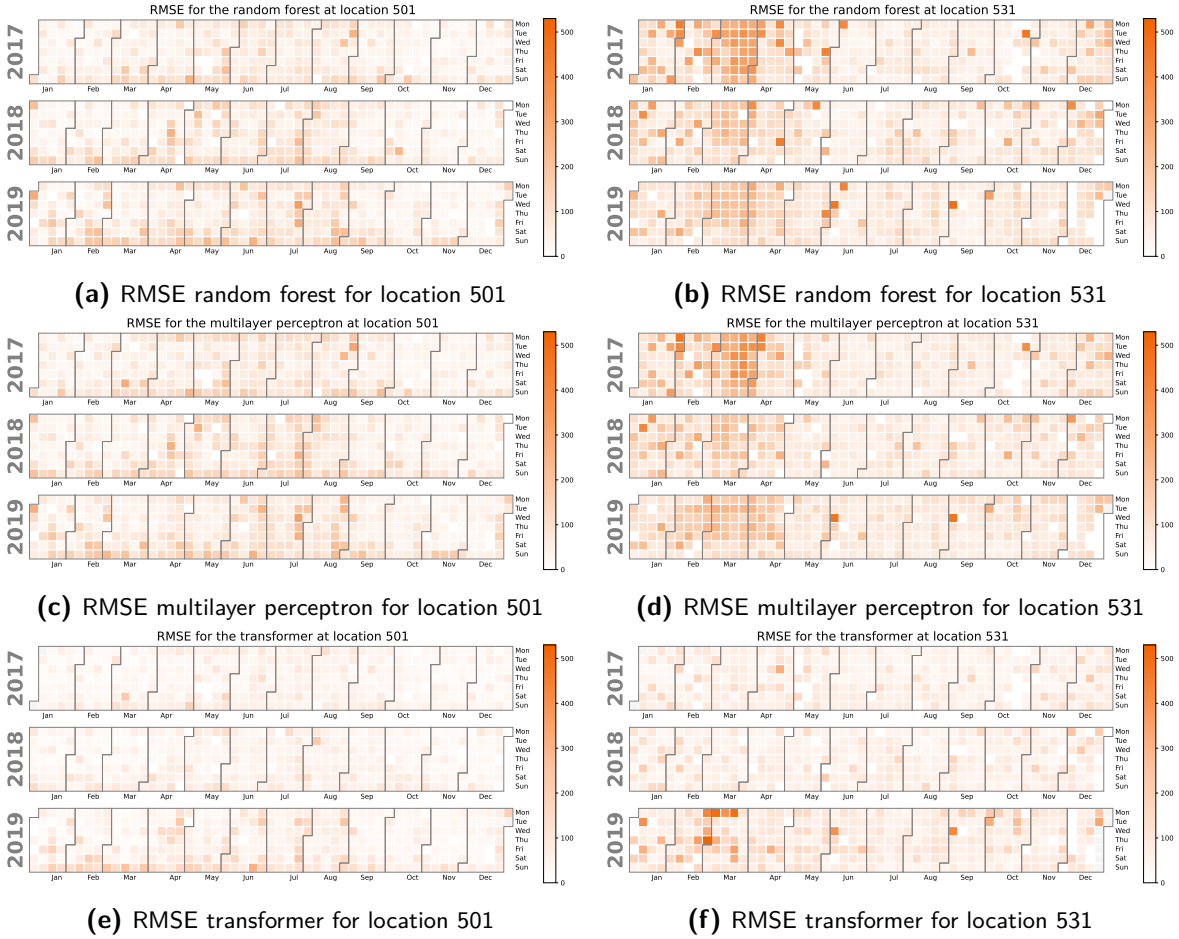
This section focuses on the performance of the prediction models throughout the year, to investigate when the prediction models encounter difficulties and whether an underlying cause can be found. Figure 7-6 shows the RMSE of each day for the different models and locations. For the transformer, it is chosen to show the multistep prediction starting at 00:00:00. However, similar results were found when looking into the errors based on a fixed prediction horizon. This can also be derived from Figure 7-4, which shows that large errors remain relatively large on all horizons. First, the results found for location 501 will be discussed. Next, there is elaborated on location 531.

The clustering analyses performed in Section 4-1 showed a clear difference in traffic behavior during summer and winter. However, no clear split between these days is present. Figure 7-6a, 7-6c, and 7-6e show that the baseline models encounter more difficulties during summer, whereas the transformer performance remains similar over the year. Moreover, two clusters were formed containing irregular days that are subject to more traffic than general and are indicated by the dark and light orange clusters. The days corresponding to these clusters are found to have the highest errors over the years. This is expected because a clear explanation for this behavior was not found. Therefore, more research should be done to investigate additional important features, such as events, that might cause this behavior. The transformer outperforms the baseline models on irregular days and during summer and winter, which indicates the adequacy of the additional information available in the transformer.

For location 531, the performance is similar throughout the year and for the different models.

In the cluster analyses the effect of school vacations was clearly shown. The figures indicate that this behavior can be predicted by all models. The performance in March is remarkably bad. The clusters show that most days in March 2017 are assigned to a separate cluster. The corresponding traffic flow shows that these days contain relatively low, or even zero traffic flow values around 17:00:00. This is implausible, however, because it is only for one or two consecutive hours, these data points were not filtered in the data analysis in Chapter 3. The baseline prediction models are unable to model this behavior and consequently encounter large errors in the train and validation set. This explains why the baseline prediction models have a better performance on the test. On the other hand, the transformer includes this behavior, which seems beneficial. However, this behavior might be caused by measurement errors. Therefore, the transformer might model erroneous behavior. As a result, the dark orange days in the test set occur because the transformer predicts zero traffic flow at 17:00:00, which causes the high RMSE shown in Figure 7-4b and the red rectangle in Figure 7-5b.

The irregular days in March 2017 are present at all day of the week (dow). However, the



(a) RMSE random forest for location 501

(b) RMSE random forest for location 531

(c) RMSE multilayer perceptron for location 501

(d) RMSE multilayer perceptron for location 531

(e) RMSE transformer for location 501

(f) RMSE transformer for location 531

Figure 7-6: Daily RMSE of the random forest, multilayer perceptron, and transformer on the entire data set for both locations. The darker orange indicates relative large errors. For the transformer the multistep prediction made at 00:00:00 each day is used. The white days correspond to the days filtered out during data analysis.

transformer only predicts zero traffic flow on Mondays in March 2019. The exact reasoning behind the transformer is unknown, which highlights the disadvantage of the model being a black-box model. However, the following is reasoned. The specific behavior occurs on consecutive days during the week but not on the weekend. The transformer knows whether this behavior occurred the previous day. However, when predicting Mondays, Saturdays and Sundays are provided in the encoder input, which do not show this behavior. This might cause the transformer to predict the low traffic flow on Mondays based on other features.

The RMSE and MAE of the prediction of 17:00:00 over multiple prediction horizons for location 531 gave contradictory results. The RMSE of the transformer was significantly higher on all prediction horizons than those of the baseline prediction models. However, the MAE of the transformer was lower on all prediction horizons. Recall, that the RMSE gives a relatively large weight to high errors, whereas the MAE weighs all errors equally. The errors, shown in Figure 7-6 highlight that the baseline prediction models encounter difficulties during the entire month of March in the test set. On the other hand, the transformer only has a few days in March on which it has a bad performance. However, these are worse than the errors made by the baseline models. This causes a significant RMSE, but a relatively low MAE.

To investigate whether the dark orange parts for location 531 are indeed caused by the data of March 2017, the baseline models and transformers are trained again but on a data set excluding this data. These models perform better, especially in March, which highlights that this data is indeed the underlying cause. If desired the corresponding graphs can be found in Appendix A-6. However, whether this traffic flow data is invalid and should be removed, or is caused by an external feature that is not implemented yet, such as construction works, is unknown. Therefore, this data should not just be removed, but for future research, the cause behind this behavior should be investigated.

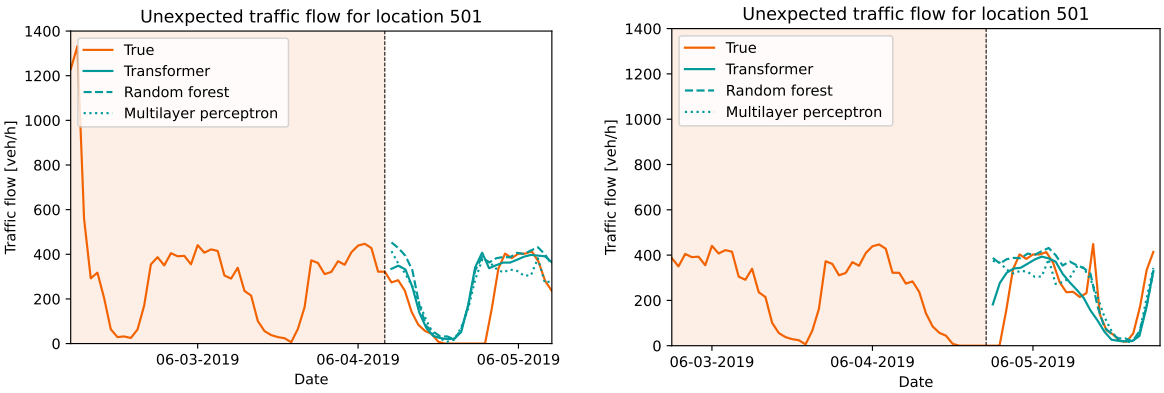
7-4 Uncertainty of the predictions

The last step in the performance characteristics is to provide an estimate of the uncertainty of the predictions. It is chosen to base this on the test set because the previous sections highlighted that the transformers are overfitting on the training data. Therefore, the performance on the test set is assumed to give a better indication of the performance during inference. However, it should be noted that the values only provide an indication and not a guarantee of the behavior of the prediction model when applied to a new data set.

The uncertainty of the predictions is investigated in three steps. First, the relative error $e_{t,h}$ at start time t and prediction horizon h is calculated as

$$e_{t,h} = \frac{1}{n} \sum_{i=1}^{n} \frac{\tilde{y}_{t,h,i} - y_{t,h,i}}{\max\left(y_{t,h,i}, 1\right)} \cdot 100\%, \tag{7-1}$$

where the maximum in the denominator is taken to ensure that the fraction cannot be undefined. Moreover, n equals the total number of evaluations corresponding to start time t and horizon h. These are investigated separately because the previous sections showed a difference in model performance during different times of the day and prediction horizons. Next, the distribution of the errors is investigated to decide how these boundaries should be derived. The relative errors contain significant positive outliers, whereas no negative outliers this size



(a) Prediction made at 06-04-2019 18:00:00

(b) Prediction made at 06-05-2019 07:00:00

Figure 7-7: (a) Shows the true and predicted traffic flow for location 501 of the largest relative error made by the transformer. In addition, in (b) the true and predicted traffic flow 13 hours later are shown, which illustrates that the transformer is able to adapt to the unexpected traffic behavior. The dotted line represents the time when the prediction is made.

are identified. The following subsection looks into these significant relative errors. Next, the distribution of the relative errors is converted to uncertainty boundaries.

7-4-1 Analyses of large relative errors

To get a grasp of what causes these large errors, the prediction made by the transformer for location 501, containing the largest relative error, is shown in Figure 7-7a. The prediction models are unable to predict the zero traffic flow and consequently, the relative error is significant. Figure 7-7b shows the prediction made 13 hours later, the baseline prediction models make the same prediction because they are subject to the same information. However, the prediction made by the transformer at the specific hour is decreased, which shows that the transformer considers the previous traffic flow and benefits from the additional information.

Figure 7-8 displays another example of a significant relative error. The prediction models are unable to anticipate the sudden drop in traffic flow. However, the timestamp corresponds to remembrance day. Therefore, it is reasonable that there is almost no traffic between 19:00:00 and 20:00:00. Nevertheless, this behavior is not included in the prediction models, since this is such a specific event.

At last, significant errors are also found at hours of the day, when the locations are subject to little traffic. As an illustration, if the traffic flow equals $1\frac{veh}{h}$ but the transformer predicts $22\frac{veh}{h}$, a large relative error is found. However, an absolute error of $21\frac{veh}{h}$, is not as unsatisfactory as the relative error suggests.

To conclude, the substantial positive relative errors, are caused by implausible and unpredictable traffic flow. In addition, large relative errors are found corresponding to reasonable absolute errors. In Chapter 3, the data analysis mainly focused on filtering out implausible peaks. However, little focus has lied on implausible low traffic flow. Therefore, additional data analyses should be implemented, without filtering out implausible but true traffic behavior.

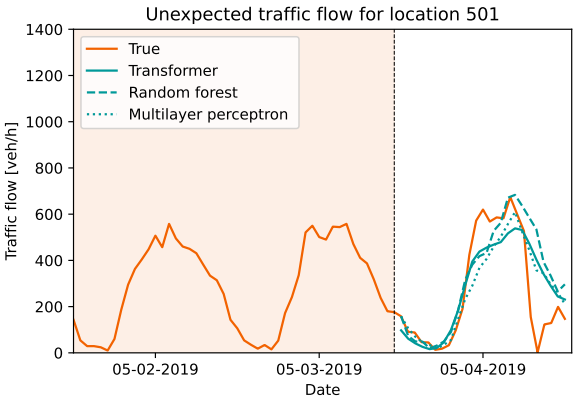


Figure 7-8: True and predicted traffic flow for location 501, containing a large relative error made by the prediction models, when the true traffic flow suddenly equals zero. The dotted line represents the time when the prediction is made.

7-4-2 Uncertainty boundaries

Due to the several positive outliers in the relative errors, it is chosen to derive uncertainty ranges by percentiles of the relative errors. The investigated percentiles (p) are 5%, 25%, 50%, 75%, and 95%, such that a boundary $\gamma_{t,h,p}$, for each p is obtained. As an illustration, if 75% of $e_{t,h} \leq 60$, then $\gamma_{t,h,75} = 60$. By rewriting (7-1), this indicates that 75% of the time $\tilde{y} \leq 1.6y$, which can be rewritten as $y \geq \frac{1}{1.6}\tilde{y}$. Therefore, these boundaries can be used to indicate the uncertainty ranges as

$$\begin{aligned} y_{t,h,p} &= \frac{1}{1 + \frac{\gamma_{t,h,p}}{100}} \tilde{y}_{t,h} \\ &= \alpha_{t,h,p} \tilde{y}_{t,h}, \end{aligned} \tag{7-2}$$

where $\alpha_{t,h,75}\tilde{y}_{t,h}$ now represents a lower boundary, of which 75% of the time $y_{t,h}$ lies above. If desired, a visual representation of these boundaries can be found in Appendix A-7.

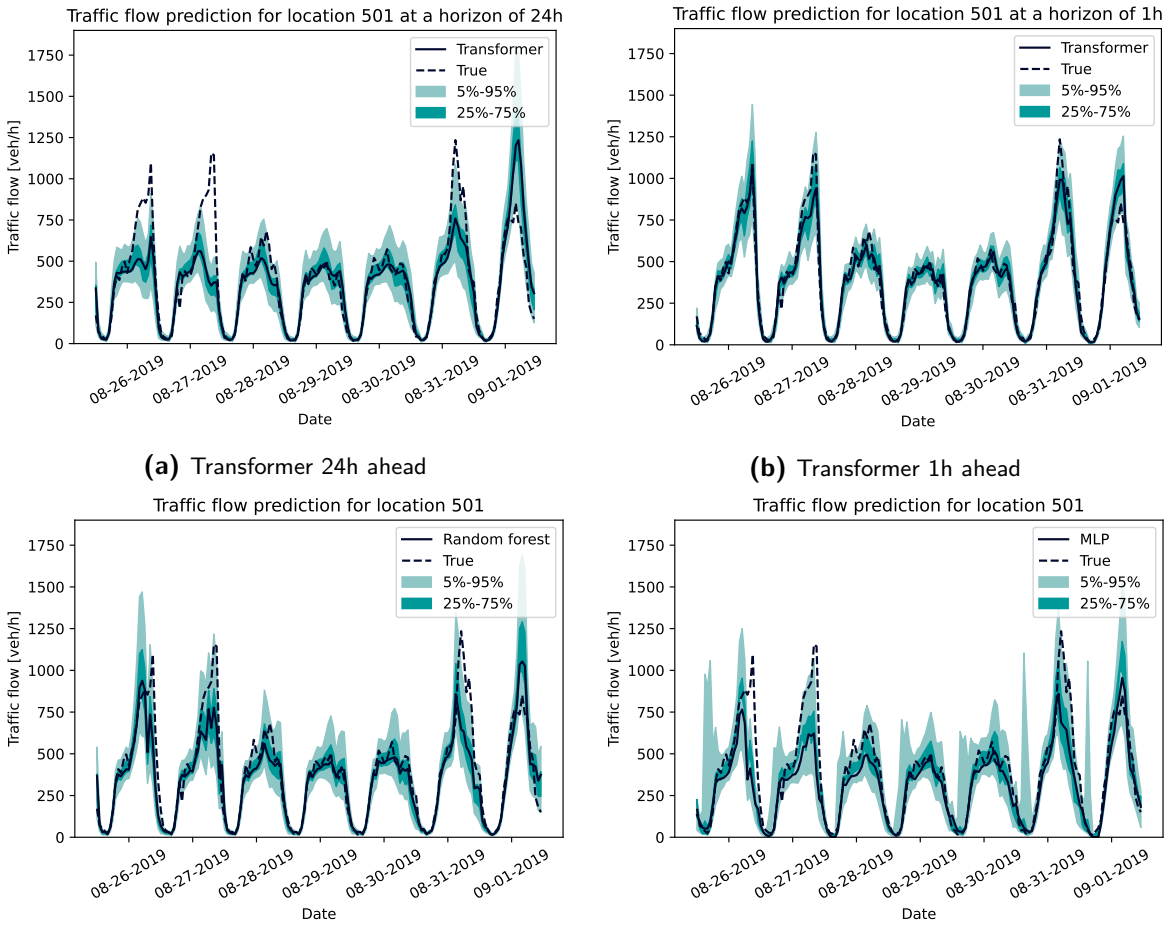
The $\alpha_{t,h,5}$ values for the MLP show two unexpected phenomena, elaborated on based on location 501. First, negative values are found at 03:00:00 and 04:00:00. Secondly, a value of approximately 14 is found at 05:00:00, which is significant compared to the other values that lie below 4. Theoretically, negative values are impossible but the predictions made by the MLP are indeed found to contain negative values. Therefore, first, setting the negative predictions to zero is examined. However, this results in two infinite $\alpha_{t,h,5}$ values, because at these two timestamps approximately 6% of the total predictions equals zero. These values are included in the boundary, which causes (7-2) to be undefined. Secondly, the negative values are set to one. Consequently, the boundaries are defined but contain large values, similarly to the value at 05:00:00. To clarify, because the traffic flow is low, a small absolute error can lead to a significant relative error that translates to a large $\alpha_{t,h,5}$ value. This second option is chosen for the comparability of the models. However, for future research constraining the output of the MLP should be investigated.

The difference in α values for the two baseline models highlights that the MLP has a larger range in uncertainty values than the random forest, which implies a small preference towards the random forest, regarding the two baseline models.

7-5 Final prediction of the transformer and baseline models

During inference, it is beneficial to apply the transformer at each hour of the day and make a multistep prediction for the next 24 hours. Next, the predictions can be updated every hour because the performance increases with a decrease in the horizon. However, for coherence, the predictions shown in this section are for a fixed prediction horizon.

To embody the final predictions, the results corresponding to two weeks in the test set for location 501 are given. First, the predictions of the last week of August are provided in Figure 7-9. This week is chosen because the clustering analysis indicates that it contains many irregular days. In Figure 7-9a and 7-9b the predictions made by the transformer at a prediction horizon of 24 and 1 hour are shown, respectively. These illustrate that at first, the transformer is not able to predict irregular behavior. However, when the prediction horizon is decreased, the performance increases significantly and the transformer can predict the irregular traffic flow. Moreover, the uncertainty ranges decrease as well. Figure 7-9c and 7-9d show the predictions made by the random forest and MLP. The random forest anticipates



(a) Transformer 24h ahead

(b) Transformer 1h ahead

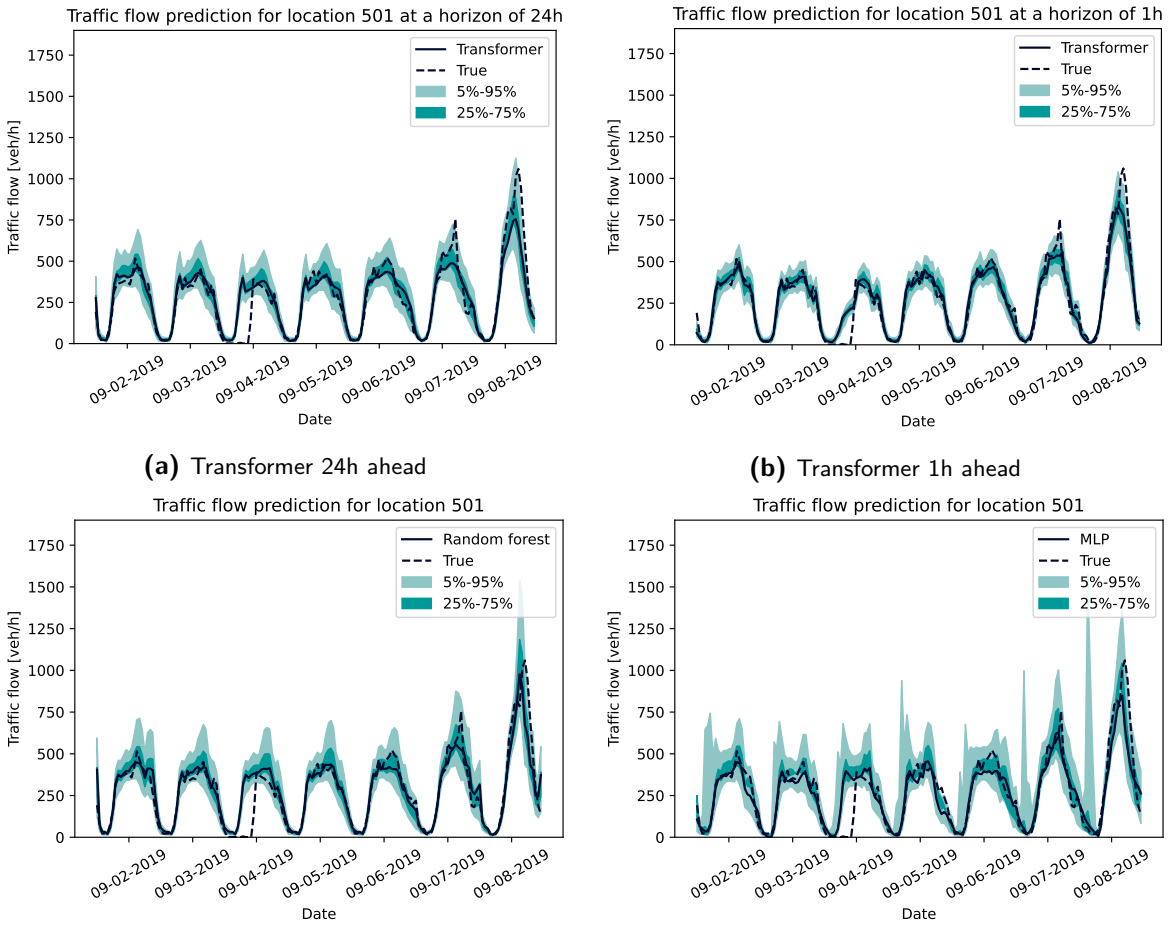
(c) Random forest

(d) Multilayer perceptron

Figure 7-9: Traffic flow predictions for the last week of August in 2019, made by the transformer at a prediction horizon of 24 and 1 hour, the random forest, and the MLP.

the irregular traffic behavior a bit and performs even slightly better than the transformer at a prediction horizon of 24 hours. As opposed to the MLP, which shows a worse performance. The uncertainty ranges of the baseline models are relatively large. In addition, strange peaks in the $5\% - 95\%$ range of the MLP occur. These are caused by the significant $\alpha_{t,h,5}$ values, as discussed in the previous section and impose difficulties when a bit of traffic flow is predicted at these hours.

The traffic flow predictions during the first week of September in 2019 are shown in Figure 7-10, which contain more regular traffic. These figures highlight that all prediction models can predict the traffic flow well. Except for a few hours at 09-04-2019 that seem to resemble an error in the data. Again, the performance of the transformer improves with the decrease in the prediction horizon, and the MLP shows high peaks in the uncertainty ranges. The examples provided both correspond to dates during summer, which are relatively difficult to predict. If desired, the predictions of a relatively easy week, the first week of January in 2019, can be found in Appendix A-8. These show a significantly better prediction with small uncertainty ranges for all prediction models.



(a) Transformer 24h ahead

(b) Transformer 1h ahead

(c) Random forest

(d) Multilayer perceptron

Figure 7-10: Traffic flow predictions for the first week of September in 2019, made by the transformer at a prediction horizon of 24 and 1 hour, the random forest, and the MLP.

7-6 Summary

This chapter evaluated and compared the performance of the prediction models. First, the performance on the train, validation, and test set was investigated. The decrease in performance of the transformer when applied to the test set is significant. The difference in performance is partly due to a discrepancy between the traffic flow at different years. However, the main cause is that randomly selecting the validation set is not representative for the test set in the transformer, due to the historical traffic flow and multistep predictions. This effect was shown to reduce by implementing a more suitable validation set.

Secondly, the performance of the models on different prediction horizons is investigated. For location 501, the transformer outperforms the baseline models on all horizons but especially on the first 6 hours. For location 531 the transformer is only advantageous up to a horizon of 3 hours. In addition, the performance on different hours of the day is investigated and different model behaviors are identified. Comparing these, in combination with the traffic flow characteristics lead to the following hypothesis. The transformer is advantageous on both short- and long-term predictions when subject to irregular traffic flow. Otherwise, the transformer is only advantageous on shorter horizons and the performance might even degenerate below the performance of the baseline models on further horizons. How to get most out of the different predictions during implementation will be discussed in the next chapter. In addition, it was shown that location 501 has many subsequent locations with a broader range of traffic flow values. Consequently, irregular behavior might be indicated earlier, underlining why for location 501 the transformer is superior for a relatively long prediction horizon.

Next, the days of the year, where the prediction models tend to be less accurate are investigated. For location 501 the transformer outperforms the baseline prediction models, throughout the entire year. The days assigned to the clusters with irregular traffic behavior contain the highest errors. For location 531, the overall performance of the models is similar but high errors are found in March. Implausible traffic flow is identified in March 2017, which causes the baseline prediction models to perform better on the test set, the significant prediction errors in March 2019 made by the transformer, and the high RMSE on all prediction horizons for 17:00:00. Whether this traffic flow data is invalid and should be removed, or is caused by an external feature that is not implemented yet, is unknown.

Finally, an estimate of the uncertainty of the predictions is made, based on the distribution of the relative error at different hours of the day and prediction horizons. Substantial relative errors are analyzed, which are caused by implausible and unpredictable traffic flow. In addition, large relative errors are found that correspond to reasonable absolute errors. Therefore, the analyses in Chapter 3 should be extended by looking into implausible low traffic flow. Next, the distribution of the relative errors is used to provide a $5\% - 95\%$ and $25\% - 75\%$ uncertainty range.

Several results of the final predictions and uncertainty ranges are provided for location 501. These illustrate that the performance of the transformer improves and the uncertainty ranges decrease, with a decrease in the prediction horizon. The random forest has a comparable behavior to the transformer on a prediction horizon of 24 hours. Moreover, the MLP has the worst performance and the highest uncertainty ranges.

Chapter 8

Conclusions and recommendations

To cope with the increasing pressure on transportation networks, more efficient use of the multi-modal transportation network should be made. This requires insights into the traffic behavior on a long prediction horizon, which highlights the necessity of traffic flow prediction models. In this thesis, I proposed a generic long-term traffic flow prediction model that can predict 24 hours ahead and that incorporates temporal and external features. To consider the sequential characteristics of traffic flow, without being subject to the limitations inherent in recurrence-based models, the transformer is implemented as the prediction model. However, whether this model applies to longer prediction horizons was not yet discussed in the literature. To synthesize this traffic flow prediction model, multiple correlation analyses are implemented to investigate important features. In addition, the adequacy of the transformer in long-term traffic flow predictions is investigated. The contributions of this MSc thesis are summarized as:

1. Insights into correlations and consequently external features that should be taken into account for long-term traffic flow predictions, are obtained.
2. A generic transformer-based model is designed that incorporates these features.
3. Insights into the applicability of the transformer on longer prediction horizons are acquired.

In Section 8-1 and Section 8-2, concluding remarks regarding these topics are made. Next, topics that lend themselves for future research are discussed in Section 8-3.

8-1 External factors in long-term traffic flow predictions

For commercial feasibility, a widely applicable prediction model is desired. Therefore, to test the genericity of the prediction models, all analyses are conducted at two locations subject to different traffic behavior. The first is located on the ring road of Haarlem and is mainly

affected by commuter traffic, whereas the second is located on the road to the coast and has more irregular behavior. These locations are referred to as location 531 and location 501, respectively.

Correlation analyses are implemented to identify important features. By clustering similar days, the day of the week (dow) and national holidays are found to be important. In addition, the season and school vacations are revealing for location 501 and location 531, respectively. Next, cross-correlation analyses between the traffic flow and weather features showed that the temperature, radiation, and relative humidity should be included. The dew temperature and sun duration are neglected due to redundancy. Equivalent results were obtained for both locations, except for the temperature that is more vital for location 501. Finally, auto-correlation analyses indicated that the current traffic flow is correlated with the previous few hours and with the traffic flow at a similar time on previous days. The difference between the locations is that for location 501, a correlation is noticed with all past days, while location 531 is mainly correlated with the same dow and neighboring days.

The differences in correlations encountered are explained by the location characteristics. Due to the commuting traffic, the traffic flow is expected to decrease during school vacations and to show a strong resemblance on the same dow, which furthermore supports that in the auto-correlation similar dows have a stronger correlation. On the other hand, it is reasonable that the road to the coast is exposed to different traffic during the summer, which is related to the temperature and additionally supports the auto-correlation with all previous days.

The relative importance of the features in the random forest indicates that the important features are consistent with the correlation analyses. On the other hand, in the transformer, some features unexpectedly appear to be less important. Due to the historical traffic flow, certain periodicity's are indirectly already included, such as the season. Consequently, these specific features do not provide additional information. Nevertheless, the application of redundant features is not detrimental to performance. Therefore, for genericity and comparability, it is chosen to implement all input features in the different models. However, for efficiency, it may be valuable to exclude some features.

8-2 Model performances in long-term traffic flow predictions

In general, for location 501, the transformer outperforms the baseline models on all prediction horizons and is especially superior at the first six hours. On the other hand, the transformer for location 531 is superior up to the first three prediction horizons. By looking at the performances of both transformers at different times of the day, combined with the deviation in traffic flow noticed at that hour, a few conclusions are drawn.

When the locations are subject to a small range of traffic flows, the performance of the models is similar and remains constant over the prediction horizons for the transformer. On the other hand, when subject to a broad range of traffic flows the performance increases as the horizon decreases and the transformer outperforms the baseline prediction models, at least for the first few horizons. This is reasonable because when subject to a small range of traffic flows, the traffic flow is mainly based on the time of the day and maybe the dow. Therefore, applying the transformer or reducing the horizon only slightly affects the prediction. On the other hand, the traffic flow is more complicated when subject to a larger range of traffic flows. This might

also be shown in the previous traffic flow, highlighting the advantage of the transformer and a decrease in the horizon. As mentioned in the previous section, location 531 mainly contains commuter traffic and consequently primarily exhibits the first type of behavior. The reason that for location 501 the transformer is superior for a relatively long prediction horizon is that the location has many consecutive hours with a broader range of traffic flow values. Consequently, irregular behavior can be indicated earlier.

This leads to the hypothesis that the adequacy of the transformer is location-dependent and promising for locations exposed to irregular traffic flow on both short and long horizons.

From the performance throughout the year for location 501, it is concluded that the transformer can identify irregular days. However, the days identified to have irregular behavior still contain the highest errors. Moreover, for location 531, implausible traffic flow is identified in March 2017, this causes significant errors made by all prediction models. However, it is unknown whether this traffic flow behavior is caused by invalid data or by unknown external features. Therefore, additional analyses need to be conducted for both locations to look at other potential input features, which will be discussed in Section 8-3.

An estimate of the uncertainty is given by looking at the distribution of the relative error at each time of the day and the prediction horizon. The significant positive relative errors are caused by little, implausible, or unpredictable traffic flows. Implausible low traffic flows were not considered during the data analysis. Therefore, additional data analyses techniques should be investigated without filtering out actual irregular behavior. This can be accomplished by incorporating additional data sources, such as data of neighboring locations or floating car data, and investigating whether the implausible data is also indicated by the additional data.

The transformer prediction improves with a decrease in the horizon. Therefore, it is recommended that multistep predictions are made every hour and updated accordingly. The overall goal is to obtain insights into traffic behavior, to provide multi-modal itineraries and influence traffic flows. The predictions made 24 hours ahead by the transformer and random forest already provide a good indication of the traffic behavior. Moreover, since most itineraries last less than a few hours, these can be updated in time, before departure. On these accounts, the transformer is considered promising for this application.

For location 501, the transformer outperformed the baseline prediction models at all prediction horizons and is consequently the recommended model. However, for location 531, the baseline prediction models are superior at further horizons. Therefore, it is suggested to make a baseline prediction with the random forest and update this prediction for a maximum prediction horizon of three hours with the transformer. Future research should explore other combinations, such as combining predictions of different models. For this, a more thorough investigation should be conducted to determine which model performs best under which conditions, such that the most suited model can be chosen or multiple models can be combined.

8-3 Recommendations

This thesis has highlighted the adequacy of external features and the transformer on long-term traffic flow predictions. Nevertheless, multiple subjects lend themselves to further research. First, adjustments to improve the implemented prediction models are discussed. Next, there

is elaborated on the extension to a multi-model prediction model. Finally, the applicability to other research fields is considered.

8-3-1 Improvements to the designed prediction models

First, the performance of the transformer degenerates significantly when applied to the test set. This is partly due to the discrepancy between years that will be less if the models are updated throughout the year, and can be achieved with offline or online methods [40]. In the latter case, the model is immediately updated when new data is available. However, this requires more resources and continuous monitoring of the data for measurement errors. Moreover, the correlation analyses showed that the traffic flow behavior does not change rapidly. Therefore, offline learning is better suited for this implementation, which can be performed periodically or when a certain value, e.g., the increase in mean error, exceeds a certain threshold. How to update the model without forgetting previously learned behavior or implementing old data that is no longer relevant is an interesting topic, of which several possibilities are nicely discussed in [44]. These range from retraining with regularization in the model, to expanding the model by additional components that are based on the new data. Many advanced techniques can be applied. Nevertheless, again, traffic behavior does not appear to change rapidly and good performance has already been achieved based on two years of data. Therefore, it is recommended to update the model on a fixed time interval, such as monthly, based on the historical data and the new data, and remove the oldest period from the data set. As the model parameters are updated, the removed data will still be included in the model and its effect will slowly disappear with the updates. In addition, there will be no need to store years of data.

Moreover, the main problem is that randomly selecting the validation set is not a proper representation for the test set. Therefore, the main issue is to search for a new validation set, such that the transformer can be designed properly and overfitting is avoided. Ideally, additional data can be implemented such that an entire year is available as the validation set. However, if this option is excluded, other options, such as subtracting every fifth week from the training set, also seem substantial.

The multilayer perceptron (MLP) occasionally predicts negative values. Therefore, the output of the MLP should be constrained, to ensure feasibility. This can be done by implementing the Rectified Linear Unit (ReLU) activation function in the output layer. Note that during the data preparation the output should then be normalized instead of standardized.

Subsequently, further extensions can be considered to expand the transformer. First, the prediction models encounter difficulties at days that were grouped as irregular days during clustering. However, no underlying cause was found for this traffic behavior. Therefore, additional correlation analyses should be performed to investigate whether this behavior might be implied by other external features, such as events or construction works. The difficulty with these irregular days is that they have a very specific behavior and only happen occasionally. Therefore, if it is desired to primarily focus on traffic during specific events. It can be beneficial to implement an entirely separate model for this behavior.

Moreover, computational constraints limited the number of past input features. The auto-correlation analyses showed that traffic flow is correlated to data beyond 48 hours ago. Recall that for location 501 correlations were found at a similar time the previous days, whereas,

for location 531, correlations were found at a similar time and dow. In addition, smaller correlations were established on traffic flow only 12 hours ago that is included. This indicates that it can be valuable to find a more ingenious input. For example, more useful data can be included without increasing the total model complexity by concatenating recent, daily, and weekly data, as done in [5, 22, 24].

In addition, traffic flow is also composed of spatial features [11, 36, 59, 65, 67, 74, 76]. Therefore, graph-based methods are commonly implemented as an extension to temporal models. In general, research showed a positive correlation between the complexity and the performance of the spatial feature prediction models; allowing for more correlations, will generally induce a better performance. However, it is hypothesized that for long-term predictions, spatial features might only be beneficial when taking a large network into account and are therefore not applicable for only predicting traffic flows.

Finally, multiple locations are implemented to investigate the genericity. From which it is reasoned that the adequacy of the transformer is location-dependent and promising on both short and long horizons for locations subject to irregular traffic. To support this hypothesis, the prediction models should be applied to additional locations based on different characteristics and if necessary, the feature set should be extended accordingly.

8-3-2 Extension to a multi-modal transportation network

The main goal is to combine multiple transportation modes into one network. Therefore, for future research, it is interesting to combine multiple transportation modes and locations into one model, such that entire traveler flows can be predicted. A few steps have to be taken, (1) prediction models for other transportation modes, such as trains, buses, bicycles, etc. have to be designed, and (2) these can be combined in an overarching framework. Moreover, this framework can be extended with additional components, such as parking availability.

This leads to many interesting research topics, e.g. to gather an overall idea into the locations that people are traveling to, and identifying external features that influence these locations. Moreover, can these external features be linked to a preference in transportation modes, and can these insights be used to identify the bottlenecks in our transportation network? These questions come together with the demand for innovative solutions from municipalities, regarding the identification of hotspots in Amsterdam and increasing the network accessibility in Flevoland.

The transformer implemented in this research is designed to process time series. Data regarding public transportation modes are often not based on fixed time intervals. However, since there is looked at an hourly interval, the exact number of travelers in a specific train or bus is not interesting. on the other hand, the hourly number of passengers arriving and departing at a location is. Therefore, these models can be approached as a time series prediction task. State-of-the-art literature often implements machine learning methods, such as the random forest, MLP, and long short term memory model, to predict the number of passengers in public transport [12, 56]. The type of prediction model suited will have to be investigated but is thought to be aligned with the prediction model found to be relevant for the traffic predictions. This is based on the hypothesis that whether a mode is composed of commuter or irregular traffic is similar for different transportation modes in a similar direction. However, this assumption will have to be investigated.

One of the objectives was to obtain a generic model. To adhere to this, it can be chosen to design different models and keep these separated, as done for each location. However, the different models can also be combined in a multi-modal transportation network, by extending to a spatio-temporal prediction model as described in the section above. This provides insides into the connections between different locations and transportation modes.

8-3-3 Applicability to other research fields

This research has shown the adequacy of the transformer on long-term predictions. The hypothesis is made that the ability to model complex behavior is especially beneficial on locations subject to irregular traffic behavior. This is aligned with locations in which auto-correlation is important and on which external features have a significant impact. Therefore, the transformer model and identification of relevant external features apply to other time series prediction tasks subject to these characteristics.

State-of-the-art literature in time series predictions focuses on among others, weather prediction, solar energy production, retail in grocery stores, and the volatility of the stock market [21, 34, 37]. In these researches, the benefit of the transformer in terms of performance, as opposed to baseline models has already been shown. However, the focus has mainly lied on short-term predictions, and often external influences are not taken into consideration. Therefore, the designed transformer-based model, incorporating external features is thought to be promising for these research fields as well.

Appendix A

Appendix

A-1 Number of trainable parameters in the transformer

This section contains the derivation behind the total number of trainable parameters in the transformer, based on variable hyperparameters. These hyperparameters are, the dimension of the input feature d_x, the number of encoder and decoder layers N, the dimension of the hidden layer in the feedforward network d_{ff}, and the output dimension d_y.

First, the number of trainable parameters in a feedforward layer is derived, because these are required in the derivations of multiple components of the transformer. A feedforward layer can be written as

$$y = X \cdot w^T + b \tag{A-1}$$

If the input X is of dimension i and the output y of dimension o, the number of trainable parameters in a feedforward layer is equal to

$$n_{\text{ff}} = i \cdot o + o \tag{A-2}$$

Feedforward layer
The feedforward layer converts the input of dimension d_x to an hidden state of dimension d_{ff}. Next, the hidden state is converted back to an output of dimension d_x, such that it can be used in further calculations. Therefore, the total number of trainable parameters in a feedforward layer equals

$$n_{\text{ff}} = d_x \cdot d_{\text{ff}} + d_{\text{ff}} + d_{\text{ff}} \cdot d_x + d_x \tag{A-3}$$

Multi-head attention block
The scaled dot product can be written as

$$\begin{aligned} y &= \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \\ &= \text{softmax}\left(\frac{\left(xW_q^T\right)\left(xW_k^T\right)^T}{\sqrt{d}}\right)\left(xW_v^T\right) \end{aligned} \tag{A-4}$$

After which the multiple outputs of different attention heads are concatenated and linearly transformed as

$$\tilde{y}_{\text{concat}} = W_0 \left(||_{k=1}^{K} \tilde{y}_{\text{head,k}} \right) + b_0 \tag{A-5}$$

When multi-head attention is applied, the dimensions in the individual heads are reduced, such that the total dimension is equal. Therefore, it can be seen that the total number of parameters in the multi-head attention block is caused by 4 feedforward calculations. Moreover, in the attention layers, the output dimension is equal to the input dimension. As a result, the total number of parameters equals

$$n_{\text{mha}} = 4 \left(d_x \cdot d_x + d_x \right) \tag{A-6}$$

Add and normalize layer
The layer normalization is calculated as

$$y = \gamma \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} + \beta, \tag{A-7}$$

where E[x] and Var[x] represent the mean and the variance of vector x, respectively. Moreover, ϵ is for numerical stability in case the denominator becomes zero. In addition, the γ and β are trainable parameters. Therefore, the number of trainable parameters in a layer normalization layer equals

$$n_{\text{ln}} = d_x + d_x \tag{A-8}$$

Transformer
The total number of trainable parameters in the transformer can be calculated as

$$n_{\text{transformer}} = n_{\text{enc}} + n_{\text{dec}} + n_{\text{output}} \tag{A-9}$$

The encoder, decoder, and final output layer are all composed of components described above. The encoder contains one multi-head attention block, two add and normalize layers, and one feedforward layer. Therefore, the total number of parameters in the encoder equals

$$\begin{aligned} n_{\text{enc}} &= N \cdot (1 \cdot n_{\text{mha}} + 2 \cdot n_{\text{ln}} + 1 \cdot n_{\text{ff}}) \\ &= N \cdot (1 \left(4 \left(d_x \cdot d_x + d_x\right)\right) + 2 \left(d_x + d_x\right) + 1 \left(d_x \cdot d_{\text{ff}} + d_{\text{ff}} + d_{\text{ff}} \cdot d_x + d_x\right)) \end{aligned} \tag{A-10}$$

Moreover, the decoder is composed of two multi-head attention blocks, three add and normalize layers, and one feedforward layer, such that

$$\begin{aligned} n_{\text{dec}} &= N \cdot (2 \cdot n_{\text{mha}} + 3 \cdot n_{\text{ln}} + 1 \cdot n_{\text{ff}}) \\ &= N \cdot (2(4(d_x \cdot d_x + d_x)) + 3(d_x + d_x) + 1(d_x \cdot d_{\text{ff}} + d_{\text{ff}} + d_{\text{ff}} \cdot d_x + d_x)) \end{aligned} \tag{A-11}$$

At last, the output of the decoder is linearly transformed to the final output dimension. For which the following number of trainable parameters are used

$$n_{\text{output}} = d_x \cdot d_y + d_y \tag{A-12}$$

A-2 Clustering analysis with an increase in the number of clusters

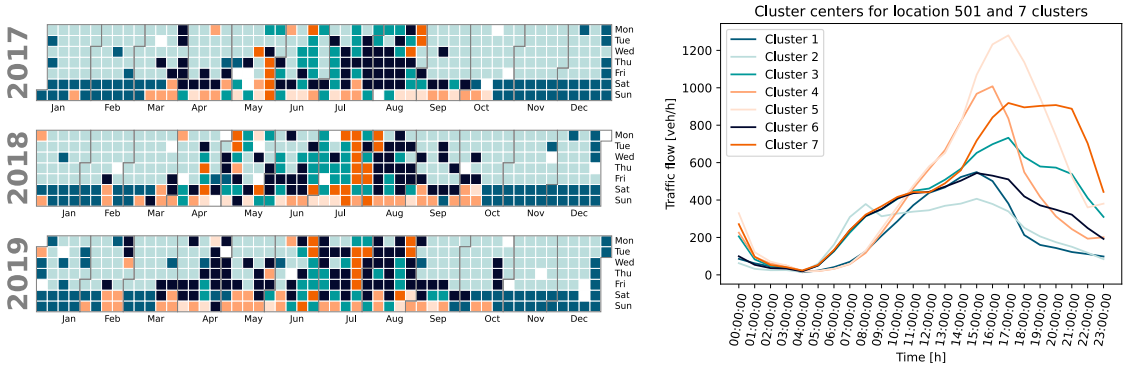
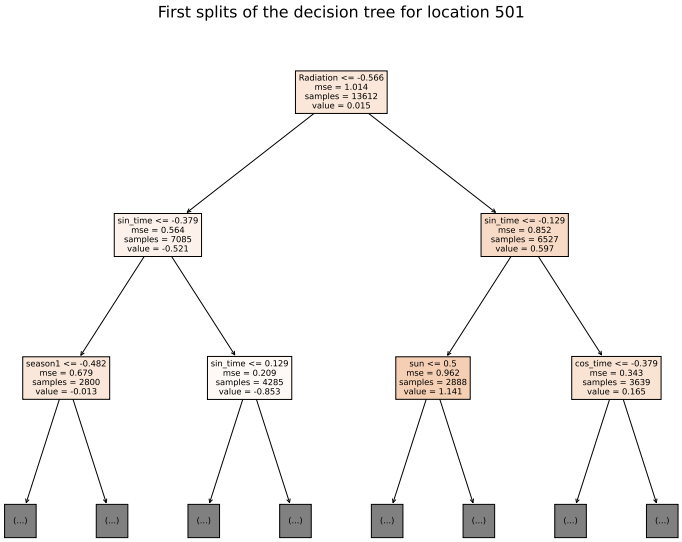
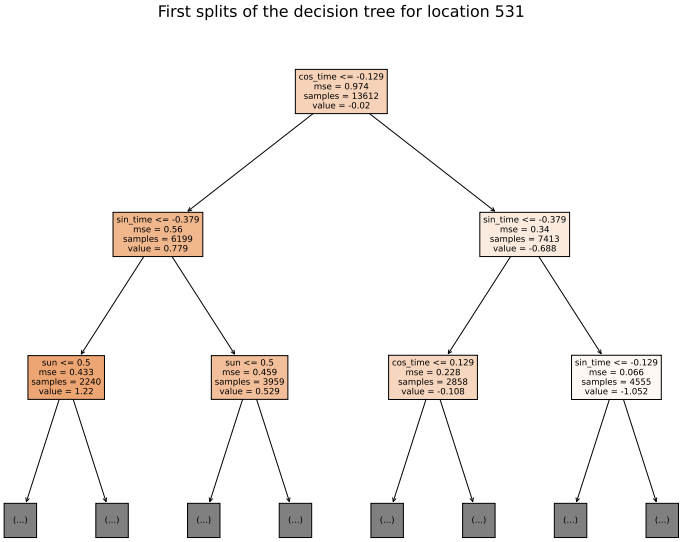


Figure A-1: Specifications of the 7 clusters obtained by hierarchical clustering for location 501. The left figure shows the division of the days corresponding to each cluster throughout the year and in the right figure the corresponding cluster centers are shown. The white days correspond to the days filtered out of the data set.

A-3 Visualization of the decision tree for location 501 and 531



(a) Location 501



(b) Location 531

Figure A-2: Decision tree shown up to a maximum depth of two for locations 501 and 531, in (a) and (b), respectively. Where the color of the nodes provides an indication of the value.

A-4 Algorithm of the transformer

Algorithm 1 Transformer implementation

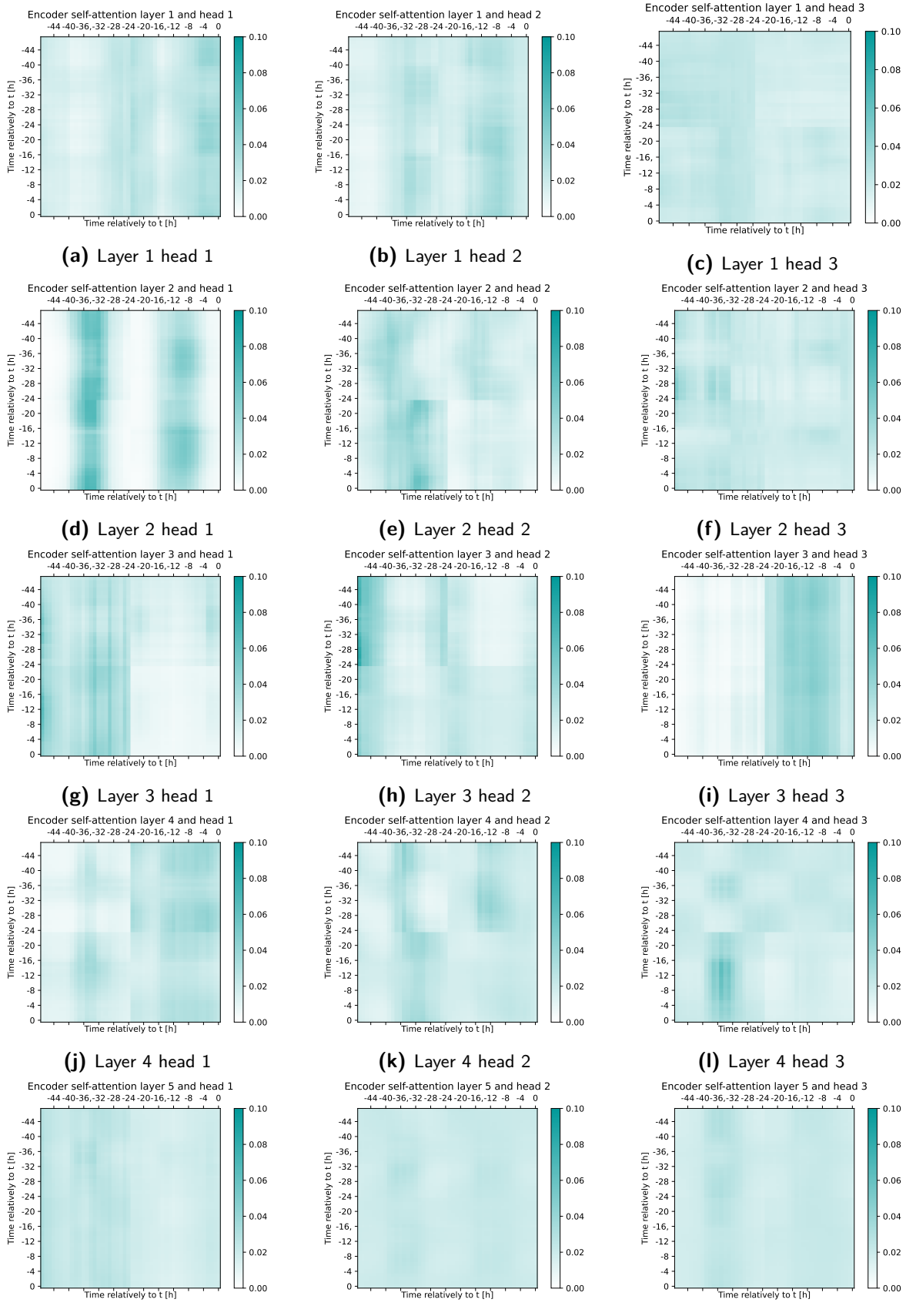
Transformer `-class(Model)`

Encoder `-class(Keras.layers.Layer)`
`for i in range(N)`
Encoder Layer `-class (Keras.layers.Layer)`
Multi-head attention `-class (Keras.layers.Layer)`
Scaled dot product attention `function`
Layer normalization `Keras.layer.Normalization`
Feedforward network `function`
Hidden layer `Keras.layer.Dense`
Output layer `Keras.layer.Dense`
Layer normalization `Keras.layer.Normalization`

Decoder `-class (Keras.layers.Layer)`
`for i in range(N)`
Decoder Layer `-class (Keras.layers.Layer)`
Multi-head attention `-class (Keras.layers.Layer)`
Scaled dot product attention `function`
Layer normalization `Keras.layer.Normalization`
Multi-head attention `-class (Keras.layers.Layer)`
Scaled dot product attention `function`
Feedforward network `function`
Hidden layer `Keras.layer.Dense`
Output layer `Keras.layer.Dense`
Layer normalization `Keras.layer.Normalization`

Final output layer `(Keras.layer.Dense)`

A-5 Insights into the transformer behavior by the self-attention weights in the encoder and decoder



(a) Layer 1 head 1 **(b)** Layer 1 head 2 **(c)** Layer 1 head 3

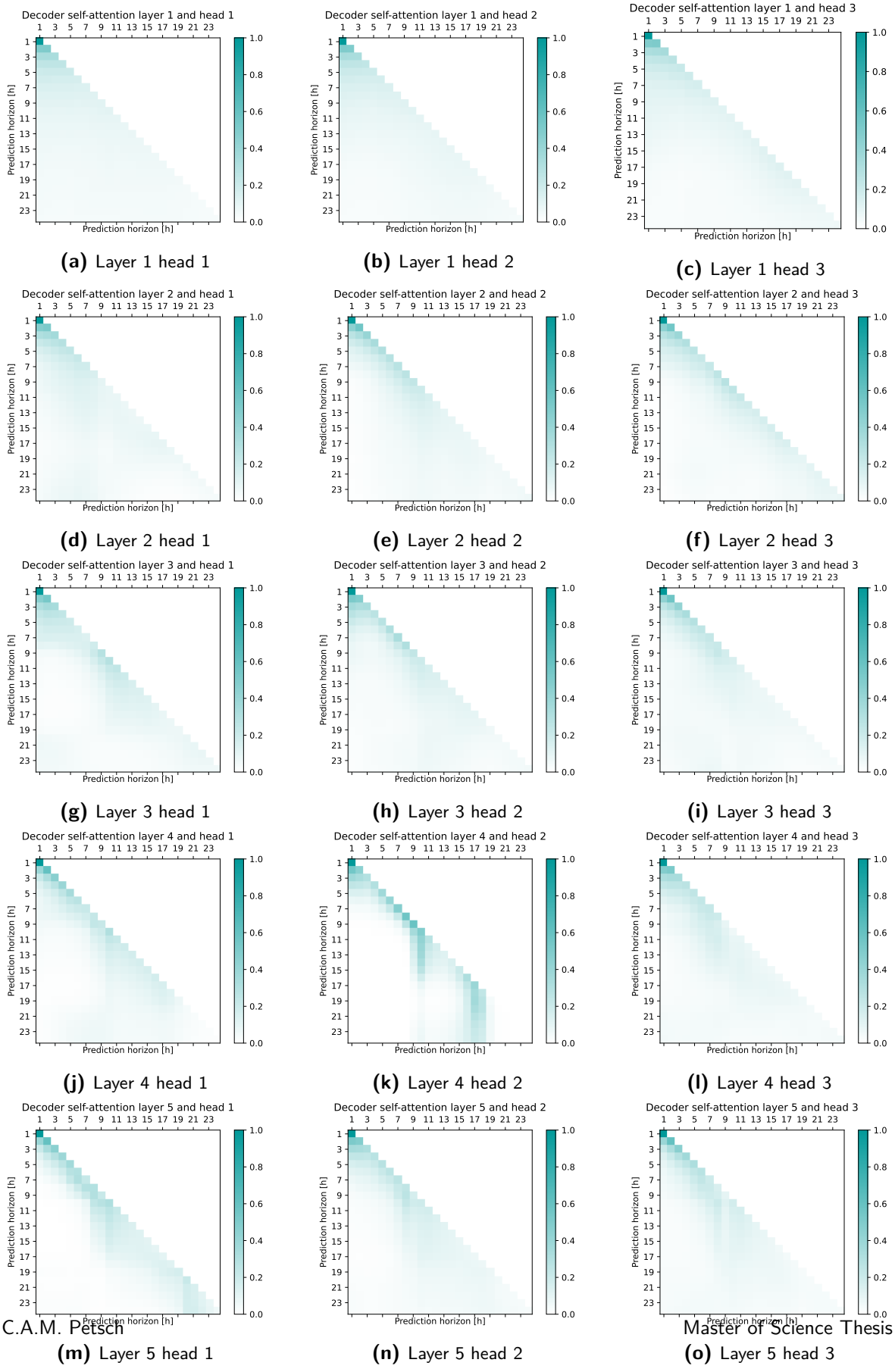
(d) Layer 2 head 1 **(e)** Layer 2 head 2 **(f)** Layer 2 head 3

(g) Layer 3 head 1 **(h)** Layer 3 head 2 **(i)** Layer 3 head 3

(j) Layer 4 head 1 **(k)** Layer 4 head 2 **(l)** Layer 4 head 3

(m) Layer 5 head 1 **(n)** Layer 5 head 2 **(o)** Layer 5 head 3

Figure A-3: Self-attention weights in the encoder for the prediction of 02-01-2019 at 00:00:00



(a) Layer 1 head 1 **(b)** Layer 1 head 2 **(c)** Layer 1 head 3

(d) Layer 2 head 1 **(e)** Layer 2 head 2 **(f)** Layer 2 head 3

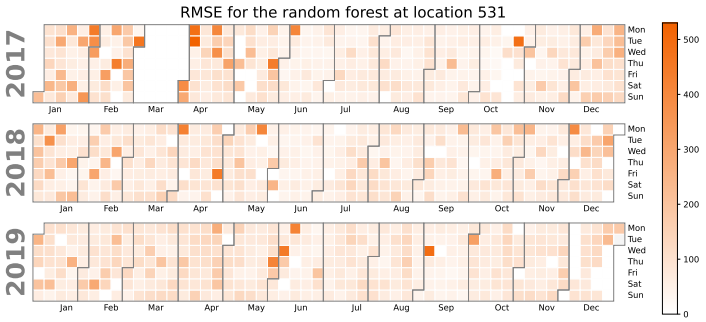
(g) Layer 3 head 1 **(h)** Layer 3 head 2 **(i)** Layer 3 head 3

(j) Layer 4 head 1 **(k)** Layer 4 head 2 **(l)** Layer 4 head 3

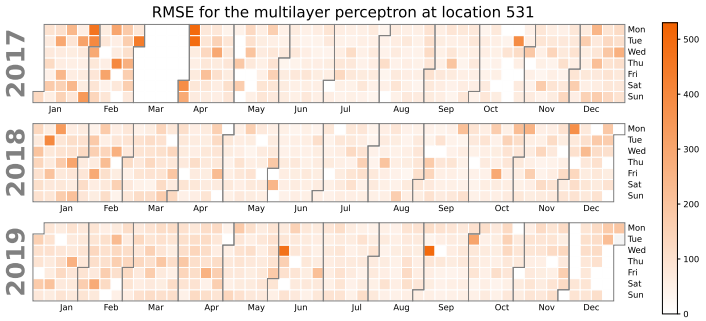
(m) Layer 5 head 1 **(n)** Layer 5 head 2 **(o)** Layer 5 head 3

Figure A-4: Self-attention weights in the decoder for the prediction of 02-01-2019 at 00:00:00

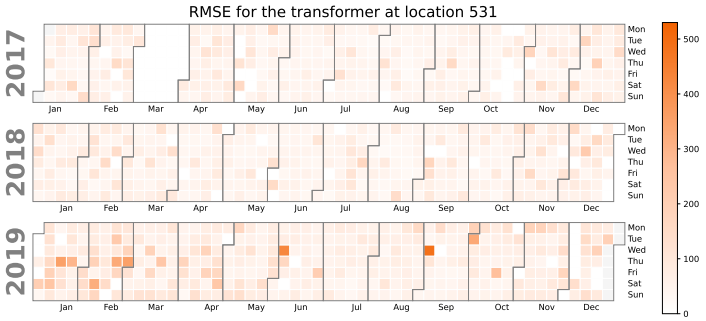
A-6 Performance throughout the year for location 531



(a) Random forest at location 531



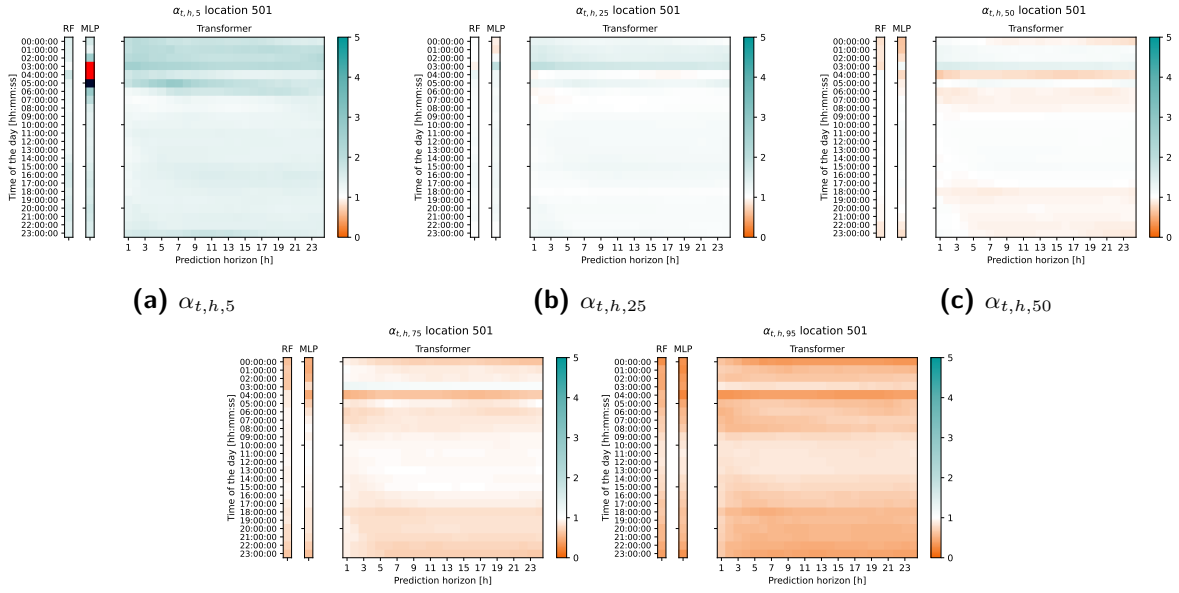
(b) Multilayer perceptron at location 531



(c) Transformer at location 531

Figure A-5: Daily root mean squared error (RMSE) of the random forest, multilayer perceptron, and transformer for location 531, without incorporating March 2017 in the train and validation set. The darker orange indicate the relatively large errors. For the transformer the multistep prediction made at 00:00:00 each day is used. The white days correspond to the days filtered out during data analysis.

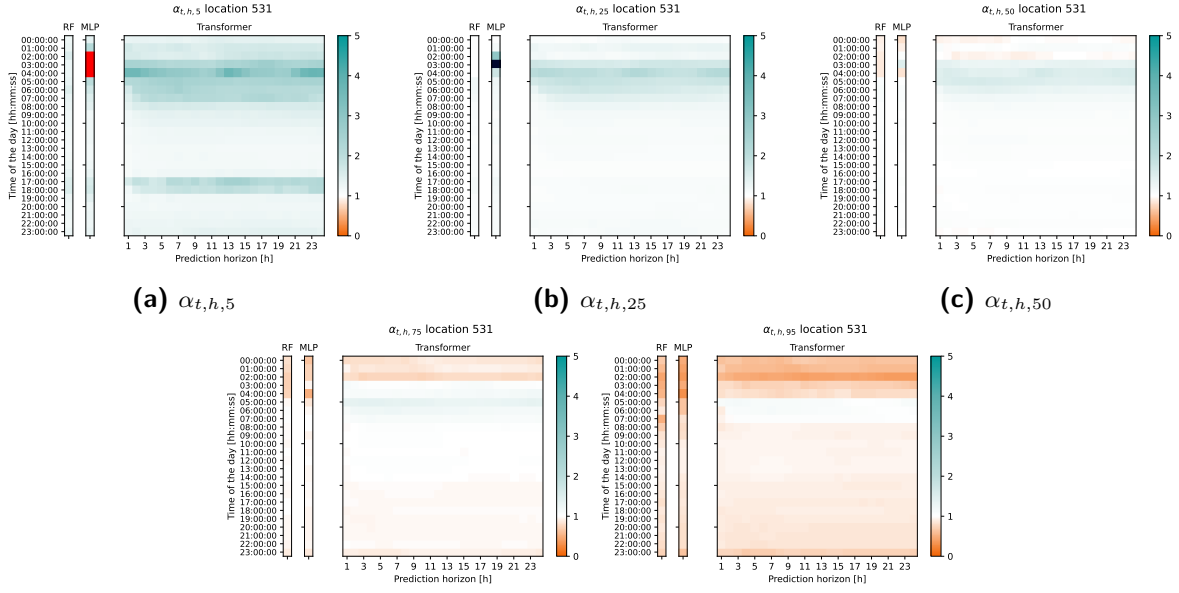
A-7 Uncertainty boundaries



(a) $\alpha_{t,h,5}$ **(b)** $\alpha_{t,h,25}$ **(c)** $\alpha_{t,h,50}$

(d) $\alpha_{t,h,75}$ **(e)** $\alpha_{t,h,95}$

Figure A-6: Uncertainty for location 501, where the orange color indicates that the boundary lies below the prediction and the blue above. The dark blue and red color represent values > 10 and < 0, respectively. For comprehensibility these are not provided in the colorbar.

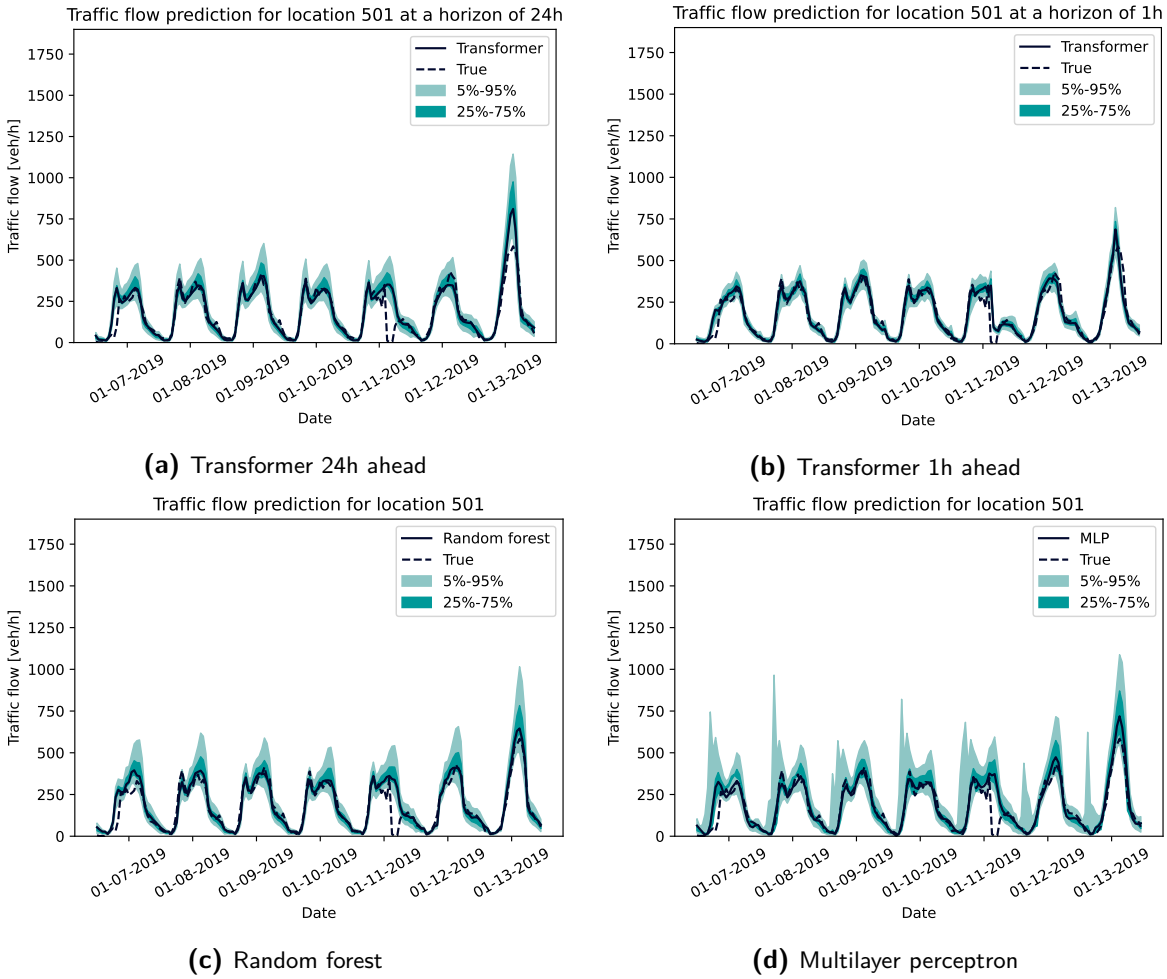


(a) $\alpha_{t,h,5}$ **(b)** $\alpha_{t,h,25}$ **(c)** $\alpha_{t,h,50}$

(d) $\alpha_{t,h,75}$ **(e)** $\alpha_{t,h,95}$

Figure A-7: Uncertainty for location 531, where the orange color indicates that the boundary lies below the prediction and the blue above. The dark blue and red color represent values > 10 and < 0, respectively. For comprehensibility these are not provided in the colorbar.

A-8 Final prediction for the first week of January 2019



(a) Transformer 24h ahead

(b) Transformer 1h ahead

(c) Random forest

(d) Multilayer perceptron

Figure A-8: Traffic flow predictions for the first week of January in 2019, made by the transformer at a prediction horizon of 24 and 1 hours, the random forest, and the multilayer perceptron. The dark blue lines show the predictions, the dotted line the true traffic flow, and the highlighted parts the uncertainty ranges.

A-9 Research paper

Long-term traffic flow predictions in a transformer-based framework

1st Carmen A.M. Petsch
Delft Center for Systems and Control
Delft University of Technology
Delft, The Netherlands
c.a.m.petsch@student.tudelft.nl

2nd Alexander Koek
Siemens Mobility
Siemens
Zoetermeer, The Netherlands
alexander.koek@siemens.com

3rd Bart De Schutter
Delft Center for Systems and Control
Delft University of Technology
Delft, The Netherlands
b.deschutter@tudelft.nl

***Abstract*—Traffic flow predictions are an important component in the rising demand for solutions to cope with the increasing pressure on transportation networks. Especially on a long prediction horizon, traffic flow predictions remain challenging due to the complex, nonlinear nature of traffic flow and the influence of both temporal and external features. To incorporate sequential behavior of time series, without being subject to limitations inherent in recurrence-based models, the transformer is increasingly used. However, whether this model is advantageous on long horizons is still unknown. Therefore, in this paper, first, multiple correlation analyses are applied to identify important features in traffic flow. Next, these are incorporated into a generic transformer-based framework, and the adequacy of the transformer on long prediction horizons is investigated based on real data. To test the genericity of the proposed prediction model, all analyses are conducted for two locations, which are subject to different traffic behavior. The first is located on the ring road of Haarlem and is mainly affected by commuter traffic, whereas the second is located on the road to the coast and has more irregular behavior. Results show that the transformer outperforms baseline prediction models on both short and long horizons, especially when the location is subject to irregular behavior. In addition, the inclusion of external features, such as the day of the week, holidays, and temperature, improves the model performance. Moreover, the genericity of the model is highlighted by its applicability to multiple locations.**

***Index Terms*—Traffic flow prediction, transformer, temporal and external features, deep learning, multistep predictions.**

I. Introduction

Traffic jams, overcrowded public transport, and limited parking availability are known daily problems in our transportation network, which cause long travel times and travelers discomfort. These issues will become even more serious as the traffic demand keeps increasing due to population growth and more intensive use of vehicles [7]. To cope with the increasing pressure on transportation networks, more efficient use of the multi-modal transportation network should be made by distributing travelers over different transportation modes, including new emerging modes based on shared vehicles. This can be achieved (1) by informing travelers about multi-modal itineraries, or (2) by influencing traffic flows. This highlights the necessity of an adequate traffic flow prediction model, such that valuable insights into traffic behavior are obtained.

Current research on traffic flow prediction often considers control of intelligent traffic systems, for which short-term predictions based on real-time measurements are required [23]. However, to inform travelers and authorities, a longer prediction horizon is required. The exact definition of short- and long-term differs throughout state-of-the-art literature, in which the division lies somewhere between a prediction horizon of 30 minutes and several hours [20, 21]. In this research, a prediction is defined as long-term when the prediction horizon exceeds one hour.

The complex, nonlinear nature of traffic flows makes it difficult to make accurate long-term predictions [24]. Therefore, the main challenge that arises with the increased prediction horizon, is the increase in uncertainty. By implementing multistep autoregressive predictions, errors propagate through the network and accumulate with time [34]. Moreover, it is difficult to make accurate long-term single-step predictions, due to the large gap in time between the prediction and the most recent available data [23]. Therefore, it is important to capture as many features describing traffic flow behavior as possible. In addition, to allow wide implementation, a broadly applicable prediction model is desired. Therefore, it is important to identify relevant external features, such that these can be incorporated into a generic prediction model.

Traffic flow is known to have a sequential behavior, which will be referred to as auto-correlation in the remainder of the paper. Advanced prediction models allow us to consider these characteristics by looking into input features of different timestamps. These methods are shown to be beneficial in terms of performance for short-term traffic predictions [13, 24]. However, whether the inclusion of auto-correlation is advantageous on longer prediction horizons is yet unknown.

Moreover, models designed to capture auto-correlation in sequential data are traditionally based on recurrent layers. However, due to the recurrent structure, these models are computationally expensive, time-invariant, and encounter difficulties with capturing long-term correlations. In the last two years, a novel prediction model known as the transformer, first proposed in [33] for natural language processing, is increasingly implemented in time series tasks. Its model structure is purely based on the attention mechanism, which allows it to focus on only the most relevant parts of a sequence, regardless of the sequential order. Consequently, the transformer is insusceptible to the limitations inherent in conventional time

series models and is often shown to be superior based on performance [22, 37].

In this paper, we propose a generic long-term traffic flow prediction model that can predict 24 hours ahead and that incorporates temporal and external features in a transformer structure. Multiple correlation analyses are implemented to investigate important features that should be incorporated in traffic flow predictions. In addition, the adequacy of the transformer in long-term traffic flow predictions is investigated. The contributions of this work are summarized as follows:

- Insights into correlations and consequently external features that should be taken into account for long-term traffic flow predictions, are obtained.
- A generic transformer-based model is designed that incorporates these external features.
- Insights into the applicability of the transformer on longer prediction horizons are provided.

The rest of this paper is structured as follows. Section II describes the state-of-the-art in this field. Subsequently, the theoretical background behind the correlation analyses and transformer framework is provided in Section III and Section IV, respectively. Next, results and analyses are shown in Section VI. Finally, conclusions are drawn in Section VII.

II. State-of-the-art in traffic flow predictions

To describe traffic flow behavior, it is important to capture both temporal and spatial features [14, 24, 41]. Temporal features describe the correlations of traffic flow throughout time and spatial features describe the correlations throughout the road network. For the scope of this research, the spatial features are excluded but they could be included in future studies, to extend the prediction model. In addition to temporal and spatial features, traffic behavior is influenced by external features such as events, the weather, and construction works [8, 9]. To identify important features, correlation analyses can be implemented [19, 38].

Various methods have been proposed for traffic flow prediction tasks. Historically, the autoregressive integrated moving average (ARIMA) and its extensions are often implemented. This is a statistical model implemented on time series and is composed of two parts, (1) the autoregressive part, which predicts the next variable based on a linear combination of previous variables, and (2) the moving average part, which takes the errors of the previous predictions into account [25]. However, the complexity of these methods is limited and nonlinear correlations cannot be captured. Therefore, these are not suited for this research [25, 39].

Recently, several authors focus on implementing machine learning methods for traffic flow predictions, which are shown to outperform the statistical prediction methods [11, 24]. Classic machine learning models implement input features corresponding to a single timestamp and include the random forest, k-nearest neighbor, multilayer perceptron (MLP), and support vector machines [6, 19, 24].

However, traffic flow is known to have a sequential behavior and models taking auto-correlation into account are shown to be beneficial in terms of performance for short-term traffic predictions [13, 24, 36]. However, it is yet unknown whether the inclusion of auto-correlation is advantageous on longer prediction horizons. Models designed to capture auto-correlation in sequential data are traditionally based on recurrent layers, which include well-known models such as the long short-term memory model and the gated recurrent unit [6, 24, 31]. However, these methods have some inherent limitations due to the recurrent structure. First, parallel computations can not be done, which makes them computationally expensive. Moreover, the models encounter difficulties with capturing long-term correlations. At last, the model dynamics are time-invariant, whereas traffic flow is not.

To address these limitations, the transformer is increasingly implemented in time series tasks and is shown to outperform the recurrence-based models [22, 37]. The transformer is insusceptible to the limitations inherent in conventional time series models because the model structure is purely based on the attention mechanism, first proposed in [1], which suggests a structure that searches for relevant input information while diminishing irrelevant information. The weight given to a specific input for a specific output is based on a similarity score between the input and output features. First, during training, all computations can be made in parallel, making it computationally more efficient. As an illustration, in [37] and [39] the training time is shown to decrease approximately 14 times when implementing a transformer instead of a gated recurrent unit. Secondly, the maximum path length between an output and an input is decreased. According to [17], the larger the path between two variables, the harder it is for the model to learn correlations. Therefore, the transformer encounters fewer difficulties regarding long-term correlations [9]. Finally, because the similarity scores are based on the input features, the model dynamics vary over time. This highlights why models based on the attention mechanism are gaining popularity in the field of traffic predictions.

III. Correlation analyses

As discussed in Section II, it is important to incorporate temporal and external features. This highlights the necessity to make a hypothesis regarding important features for the prediction model. In state-of-the-art literature, two different kinds of correlation analyses are performed for this purpose. First, clustering techniques are applied to look into the similarities between days [30, 38]. Secondly, cross-correlation techniques are implemented to investigate the correlation between two different data sets [19, 38]. Moreover, these researches imply that correlation analyses are location-specific and should be reconsidered when the prediction model is applied to a different location.

A. Clustering techniques

Two main methods, often implemented in state-of-the-art literature to find similarities in traffic flow behavior are k-means clustering [38] and agglomerative hierarchical clustering [30, 38]. Both methods are unsupervised learning algo-

rithms that group data based on feature similarity. However, the first classify a data set in a specified number of clusters k. Whereas the latter, creates a set of nested clusters in a tree-like structure, without specifying the number of clusters.

The disadvantages of k-means clustering are threefold. First, it requires the specification of the number of clusters beforehand. However, because clustering is implemented to analyze the data, this number is unknown. Secondly, it is required to run the algorithm multiple times, because it is sensitive to local minima. Thirdly, the algorithm experiences difficulties with clusters of different sizes, or non-spherical shapes. On the contrary, agglomerative hierarchical clustering is computationally more expensive, but it is not subject to the limitations inherent in k-means clustering. Therefore, agglomerative hierarchical clustering is found to be appropriate for this research.

Agglomerative hierarchical clustering is a bottom-up approach. It starts by assigning each day to a separate cluster. Next, the closest two clusters are identified and merged. This continues until all days belong to the same cluster. The algorithm contains two, to be specified, metrics; the distance metric and the linkage criteria. The distance metric is used to calculate the similarity between two clusters and is chosen as the Euclidean distance. The linkage criterion defines how the distance is computed, for which Ward's linkage is selected because it minimizes the total within-cluster variance and tends to avoid small-sized clusters [35]. Ward's method calculates the increase in cluster variance by merging cluster A and B as

$$I_{AB} = \frac{n_a n_b}{n_a + n_b} \left(\bar{y}_a - \bar{y}_b\right)^2, \tag{1}$$

where n_a and n_b are the number of data points in cluster A and B, respectively. In addition, $\bar{y}_a$ and $\bar{y}_b$ are the cluster centers. The increase in cluster variance is calculated for all clusters. Subsequently, the cluster with the smallest increase in squared distance is merged and the cycle continues. In the end, the algorithm outputs a dendrogram, which indicates the hierarchical relation between the clusters and illustrates the effect of an increase and decrease in the cluster granularity level. By slicing the dendrogram horizontally, the clusters are formed based on the corresponding number of clusters.

B. Cross-correlation techniques

To investigate the relationship between two data sets, cross-correlation techniques can be applied. Two methods often implemented are: Pearson's method [38], and Spearman's rank [19], which find a linear and a nonlinear correlation, respectively.

1) Pearson's correlation method: The correlation between data set $x = \{x_i\}_{i=1}^n$ and $y = \{y_i\}_{i=1}^n$, can be found by fitting a linear line through the data. The strength of the correlation is represented by Pearson's correlation coefficient (r_{pearson}), which indicates the deviation between the data and the fitted line [28]. The correlation coefficient lies between -1 and 1, where -1 denotes a strong negative correlation and 1 a strong positive correlation. Pearson's correlation coefficient is calculated by dividing the covariance of x and y by the standard deviations s_x and s_y as

$$r_{\text{pearson}}\left(x, y\right) = \frac{\text{cov}\left(x, y\right)}{s_x s_y} \tag{2}$$

2) Spearman's rank correlation method: Spearman's rank builds further upon Pearson's method. It finds a nonlinear correlation, under the assumption that the data is monotonic. This means that when one variable increases the other either never decreases or increases. In addition, Spearman's rank coefficient also varies between -1 and 1. First, the rank of each data point $x_{\text{r},i}$ corresponding to x_i is determined, which varies between 0 and 1. Then, instead of the actual values, the correlation of the ranked data set is investigated, and the correlation coefficient is calculated as

$$r_{\text{spearman}}\left(x, y\right) = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n\left(n^2 - 1\right)}, \tag{3}$$

where $d_i^2 = (x_{\text{r},i} - y_{\text{r},i})^2$ equals the squared distance between the rank of two data points [32].

IV. Transformer in traffic flow predictions

The transformer is increasingly implemented in time series prediction tasks [9, 22, 37]. This section will elaborate on the model architecture and the implementation of transformers.

A. Encoder-decoder structure

The transformer is based on the encoder-decoder structure [4]. This structure is implemented to map an input sequence to an output sequence of a different length and is composed of two components, as shown in Fig. 1. At time step t, the encoder converts the encoder inputs $x_{\text{enctot},t} = [x_{\text{enc},t-l}^T \dots x_{\text{enc},t}^T]^T$, where l denotes the number of previous steps taken into account by the model. In addition, each input is composed of the traffic flow y and additional input features x, corresponding to the specific time stamp and location as

$$x_{\text{enc},t-l} = \begin{bmatrix} y_{t-l} & x_{t-l}^T \end{bmatrix}^T \tag{4}$$

Next, the decoder uses the encoder output and decoder inputs, to predict $\tilde{y}_{\text{tot},t} = [\tilde{y}_{t+1} \dots \tilde{y}_{t+h}]^T$, where h equals the maximum prediction horizon and $\tilde{y}_{t+i}$ the traffic flow prediction at $t+i$ for $i = 1, \dots, h$. The advantage of this prediction structure is that the additional input features, corresponding to future timestamps, can be taken into account through the decoder inputs, which are set up as

$$x_{\text{dec},t+h} = \begin{bmatrix} \tilde{y}_{t+h-1} & x_{t+h}^T \end{bmatrix}^T \tag{5}$$

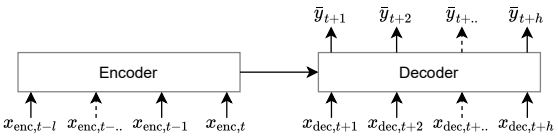


Fig. 1: Schematic representation of the encoder-decoder.

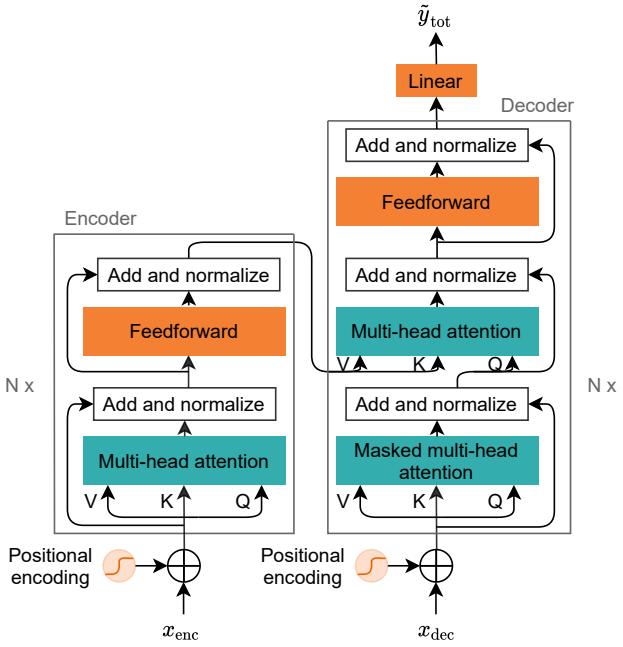


Fig. 2: Schematic representation of the transformer structure.

Here the previously predicted traffic flow is inserted because future traffic flow is unknown. How these inputs are converted, depends on the model structures used inside the encoder and decoder. For the transformer, these are based on the attention mechanism.

B. Transformer architecture

The structure of the transformer, as proposed in [33], is shown in Fig. 2. Notable are the (masked) multi-head attention blocks, the positional encoding, the add and normalize layers, and the feedforward layers, which will all be shortly elaborated on below. Each encoder and decoder layer can be stacked N times to model more complex behavior.

1) (Masked) multi-head attention: The main component of the transformer is the multi-head attention block, which is used in three different places. First, self-attention is implemented in the encoder. In addition, masked self-attention is implemented in the decoder. At last, attention is implemented between the encoder and decoder.

The idea of attention is to represent an input by a weighted sum of a sequence, to include relevant information of the entire sequence in the specific input. How much attention is given to input j to represent input i, is specified by the attention weight $\alpha_{i,j}$, which is calculated by taking the softmax of an attention score $e_{i,j}$. The softmax scales the values relative to the entire sequence value, such that these lie in a range of 0 and 1 and additionally add up to 1. Two kinds of scoring functions are commonly used in literature, additive scoring [1] and scaled dot scoring [33]. The disadvantage of the first is that it requires relatively many computations to compute the attention weights. Therefore, we only elaborated on the scaled dot scoring function.

In self-attention, the attention score is based on one specific input vector, whereas for the attention between the encoder and decoder, the attention score is based on two different vectors. However, the working principles are equivalent. The input vector is used in three ways. By comparing $x_{1,i}$ to another input $x_{2,j}$, a weight is established for output y_i and y_j. In addition, the output vector is calculated as a weighted sum of the input vector. These are called the query (q), key (k), and value (v), respectively. The input vector is linearly transformed by the trainable matrices W_q, W_k, and W_v, into dimension d, to obtain these vectors as

$$q_i = W_q x_{1,i}, \quad k_i = W_k x_{2,i}, \quad v_i = W_v x_{2,i}, \tag{6}$$

where for self-attention $x_1 = x_2$. Subsequently, the similarity measure is taken, to calculate the attention score between the query and the key, such that

$$\alpha_{i,j} = \text{softmax}\left(\frac{q_i k_j^T}{\sqrt{d}}\right), \tag{7}$$

and the output is calculated as

$$\tilde{y}_i = \sum_{j=1}^{l_x} \alpha_{i,j} v_j, \tag{8}$$

where l_x is equal to the length of the value vector. The scaling factor $\frac{1}{\sqrt{d}}$ is applied to ensure that the dot product does not go into regions where the softmax has small gradients [33]. The advantage of this approach is that matrix multiplications can be implemented. As a result, the output can be calculated as

$$\tilde{y} = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V \tag{9}$$

In the decoder, leftward information should be prevented, because future traffic flow inputs are unknown. All future inputs are therefore set to $-\infty$ inside the softmax function. Consequently, these inputs will vanish, which is referred to as masked attention.

Instead of transforming the input sequence once, in [33] it is found that it is more efficient to project the query, keys, and values multiple times into a different dimension. Next, each projection goes through an attention mechanism in parallel. Subsequently, the independent outputs are concatenated and again linearly transformed into a final output, which is calculated as

$$\tilde{y}_{\text{concat}} = W_0 \left(||_{k=1}^{K} \tilde{y}_{\text{head},k}\right) + b_0, \tag{10}$$

in which $||_{k=1}^{K}$ represents the concatenation of the K output sequences of $\tilde{y}_{\text{head},k}$. In addition, the concatenated vector is linearly projected by trainable matrices W_0 and b_0. The output of each head is based on a different query, key, and value. To ensure that the computational cost of the multiple heads is similar to that of a single head, the dimension of the query, key, and value is decreased to $\frac{d}{K}$. Consequently, the total number of trainable parameters is similar to that of a single attention head [33]. The idea behind this approach is that

multiple attention mechanisms allow for different projections, and other dependencies to be captured, such as short- or long-term dependencies, without increasing the complexity [9].

2) Positional encoding: By omitting the recurrence in the network, all information regarding the order of the input sequence is lost. This is undesired because the sequence order still contains important information. Therefore, positional encoding is implemented, which includes information regarding the relative position of the data in the input. In the original paper [33], sine and cosine functions of different frequencies are added to each input dimension, such that close data points differ in higher frequencies [15, 26]. However, the positional encoding requirements for time series predictions differ from natural language processing.

The objective is to provide information about the relative position. The simplest approach is to concatenate a vector ranging from 0-1 to the input. In this research, the number of input features taken into account is fixed. Therefore, concatenating this vector, referred to as the age feature, is sufficient to represent the relative order of the input [3, 22]. On the other hand, this method is unfeasible in natural language processing because the input dimension varies.

On top of the relative position, additional global positions are important in time series predictions, such as the month of the year or the day of the week (dow). Therefore, some researchers concatenate additional trainable layers, which learn periodicity in the data. Subsequently, these are inputted to the prediction model [22, 37]. However, this requires additional trainable model parameters and can give rise to inserting non-intuitive periodicity. Moreover, important periodicity can also be identified in correlation analyses. Therefore, it is chosen to only concatenate the age vector during positional encoding and investigate the additional global positions in the correlation analyses.

3) Add and normalize layers: Residual connections are implemented by adding the output of the attention layer or feedforward layer and the original input, after which the result is normalized. As a result, gradients are allowed to flow directly through the network, which improves the training ability [16].

4) Feedforward layers: The feedforward layer linearly projects its input two times, with a rectified linear unit (ReLU) activation function in between, which outputs the input if positive and zero otherwise. The parameters for this projection are optimized during training, to process the output of an attention layer such that it is more suitable as an input for the next attention layer. At last, a single linear layer is implemented which transforms the decoder output to the desired output dimension.

C. Implementation of the transformer

As mentioned in Section II one of the advantages of the transformer is that training can be done in parallel, which significantly decreases the computational complexity. In the encoder, all required inputs are known from the start. However, in the decoder, the previously predicted output is used to compute the consecutive output. To allow for parallel computations in the decoder, teacher forcing is applied [12]. Instead of feeding the predicted output to the next decoder input, the model then provides the known output from the training set. Therefore, the gradient of the loss function can be computed separately for each layer.

V. Experimental settings

We have selected the area of Haarlem, Bloemendaal, and Zandvoort in the Netherlands (see Fig. 3). The blue circles indicate the locations of the available detectors in this area. The data is provided by the Nationaal Dataportaal Wegverkeer (NDW)[1] and contains traffic flow data at a 5min interval for 2017, 2018, and 2019. In addition, weather data is provided by the Koninklijk Nederlands Meteorologisch Instituut (KNMI), which possesses 48 automatic weather stations in the Netherlands

The traffic flow data is processed to limit the amount of invalid data input to the models. First, the misalignment due to summer and winter time is modified. Next, two quality checks are performed to identify erroneous and inaccurate traffic data. The first check compares the traffic flow with the theoretical maximum, in order to remove the implausible data points. The second check looks into measurement errors inside the plausible ranges, by comparing irregular data points with their neighbors. Finally, missing data and implausible zero data are identified. The weather data is not processed here because it has already been filtered by the KNMI, which is assumed to be sufficient. At last, all input features are scaled by standardization, to ensure that the different features lie in the same range [24, 36, 37].

The maximum prediction horizon is set to 24 hours, divided into slots of one hour, such that the prediction of the next day is composed of 24 predictions. Therefore, the data is aggregated to an hourly interval.

Location 501 and 531, are representative of the traffic behavior of the different locations. The first lies on the road towards to coast and is mainly subject to irregular traffic behavior. The second is mainly affected by commuter traffic. Therefore, in the remainder, the focus lies on these two locations.

A. Correlation analyses

Agglomerative hierarchical clustering is implemented with the publicly available *sklearn.cluster.AgglomerativeClustering* library[2]. Based on the dendrogram a rational number of clusters is chosen to be investigated.

To identify important and redundant weather features, cross-correlation analyses are performed. Both Pearson's and Spearman's rank methods are implemented, where the latter was able to find higher correlations. Therefore, in line with the research done in [19], the second method is preferred and

[1] http://opendata.ndw.nu/

[2] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

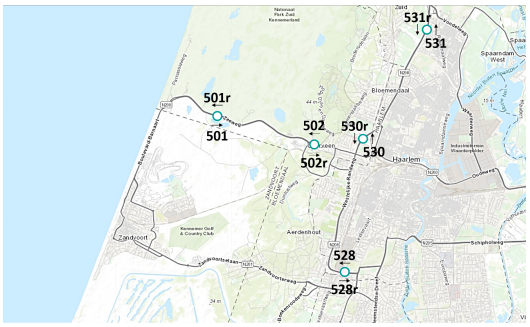


Fig. 3: Roadmap of the area of Haarlem, Bloemendaal, and Zandvoort, in which the blue circles indicate the locations of the sensors.

chosen to be elaborated on. Moreover, by plotting the data the monotonic assumption is found to hold.

The auto-correlation in traffic flow is investigated to explore the historical traffic flow measurements that should be taken into account. In line with [19], the data is detrended to remove fluctuations caused by the hour of the day and dow. As a result, it is more straightforward to see whether correlations are caused by other factors. Therefore, the weekly median is subtracted from the original traffic flow. Next, the traffic flow is shifted, such that a maximum correlation of three weeks ago can be investigated. This limit is set due to computational constraints.

B. Prediction models

A random forest and MLP are implemented as the baseline models to compare the transformer with because these are often implemented in state-of-the-art literature [6, 19, 24, 27]. The transformer and MLP are implemented in python, with the publicly available `Keras.Model` library [5]. In addition, the publicly available `sklearn.Ensemble.RandomForestRegressor` library [29], is used to implement the random forest. The code constructed to build, train, and evaluate the models is available at `https://github.com/carmenpetsch/Transformer.git`.

Bayesian hyperparameter optimization based on the tree parzen estimator is performed for the baseline prediction models and transformers [2]. After 100 evaluations the parameters are investigated and tuned a bit further. The final hyperparameters for the transformer are given in Table I. For the random forest, the number of estimators, maximum depth, and minimum samples per leaf are set to 400, 10, and 8 for location 501, and 400, 10, and 20 for location 531. The number of layers, neurons per layer, learning rate, and batch size of the MLP are set to 55, 2, 0.0001, and 16 for location 501 and 70, 2, 0.0001, and 16 for location 531. Location 531 is subject to relatively few irregular traffic flow because it is mainly composed of commuter traffic. This supports why relatively simple models are implemented for this location. The optimization algorithm used for the transformer and MLP is Adam, because it is the state-of-the-art [18]. The root mean squared error (RMSE) is implemented as the loss function, because it is desired to punish large errors [3, 10, 13, 15].

TABLE I: Hyperparameters for the transformer

	n_{heads}	Layers	Neurons	learning rate	Batch size	Dropout rate
501	3	5	360	0.0001	16	0.2
531	3	4	340	0.0001	16	0.1

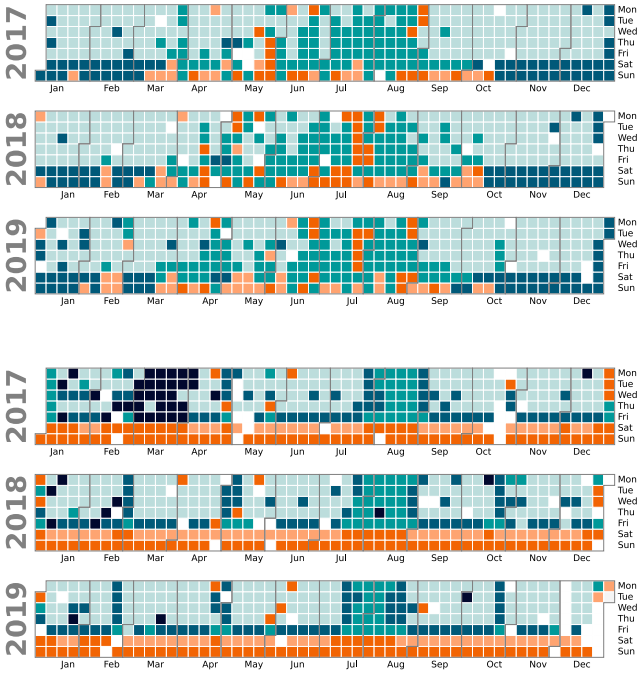


Fig. 4: Clustered days for location 501 and 531 in the top and bottom figure, respectively.

VI. Results and Analyses

The results are divided into two aspects; the identification of important features by correlation analyses and the performance of the prediction models.

A. Correlation analyses

This section investigates the correlations in the data set, to gain insights into the relevant features.

1) Daily clustering: The dendrograms retrieved by clustering for locations 501 and 531 indicate that, based on the minimum increase in cluster dissimilarity, a logical number of clusters equals 5 and 6, respectively. The clusters obtained for a lower increase in cluster dissimilarity resulted in numerous clusters containing only a few irregular days, to which no intuitive features could be assigned. Therefore, the division into more clusters is not further investigated. The clustered days are shown in Fig. 4. These show that the dow and national holidays are important. In addition, the season and school vacations are revealing for locations 501 and 531, respectively. This indicates that there is a clear difference in important

features for the locations. The light and dark orange clusters for location 501 highlight irregular traffic behavior, when the location is subject to significantly more traffic.

2) Weather and traffic flow cross-correlation: The correlation matrix for location 501 is shown in Fig. 5, in which dark green means a strong positive correlation, dark orange a strong negative correlation, and white no correlation. This indicates that the temperature, dew temperature, sun duration, and radiation are positively correlated to the traffic flow. In addition, the relative humidity is negatively correlated.

Redundant features are identified as features, excluding the traffic flow, that are highly correlated to each other. These features do not both have to be input into the prediction model, because they contain similar information. The correlation matrix indicates that the temperature and dew temperature, the sun duration and radiation, the precipitation duration and precipitation sum, and at last the wind features are redundant. Therefore, the weather features that are input to the prediction models for these locations are temperature, radiation, and relative humidity.

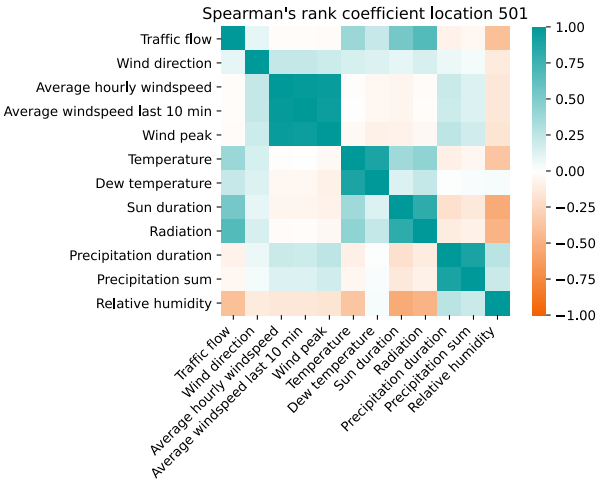
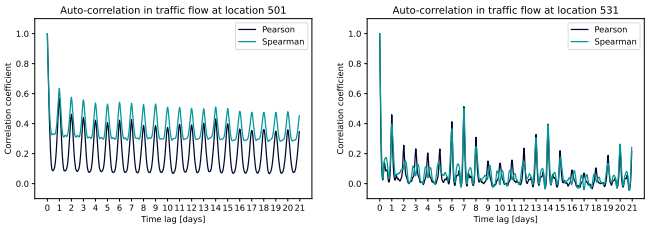


Fig. 5: Spearman's rank correlation coefficient at location 501.

Notable is that the precipitation does not seem correlated with the traffic flow. This is unexpected, both intuitively and based on state-of-the-art literature [19, 40]. This might be because weather stations are not located at the exact location of the sensors, which does not seem like an issue for the other features. However, precipitation is relatively more location-specific than the other weather features. Moreover, the only difference in correlation coefficients between the two locations is noticed in the temperature and dew temperature, which are more important for location 501.

3) Auto-correlation in traffic flow: The auto-correlation at locations 501 and 531, are shown in Fig. 6a and Fig. 6b, respectively, which indicate a clear difference in traffic behavior between the two locations. Location 501 correlates with all previous days, at approximately the same hour of the day. Moreover, a slightly higher correlation is noticed for the same



(a) Location 501 (b) Location 531

Fig. 6: Auto-correlation of the traffic flow for 3 consecutive weeks, without the influence of the hour of the day and dow.

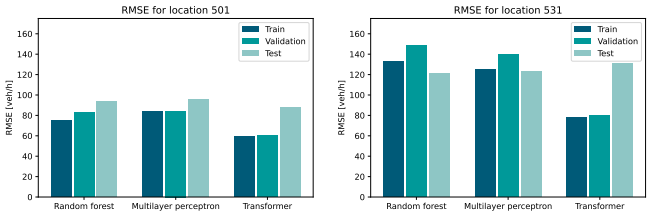
dow a week ago. On the contrary, location 531 shows a high correlation with the same and neighboring days at a similar time one, two, and three weeks ago. However, a negligible correlation is found with the other previous days. Moreover, the correlation pattern for both locations decreases as the prediction horizon increases. It is chosen to limit the number of previous measurements taken into account to 48 hours due to computational constraints. However, it can be beneficial to extend the input data with the traffic flow at the same hour a week ago, as done in [3, 13, 15].

The differences in correlations for the locations can be explained by the location characteristics. Location 531 is located on the ring road and is mainly affected by commuter traffic. Whereas location 501 is located on the road to the coast and has more irregular behavior. Due to the commuter traffic, the traffic is expected to decrease during school vacations and to show a strong similarity between the dow, which supports that in the auto-correlation similar dow have a stronger correlation. Furthermore, it is reasonable to assume that the road to the coast is subject to different traffic during the summer, which is related to the temperature and supports the auto-correlation with all previous days.

B. Long-term traffic flow predictions

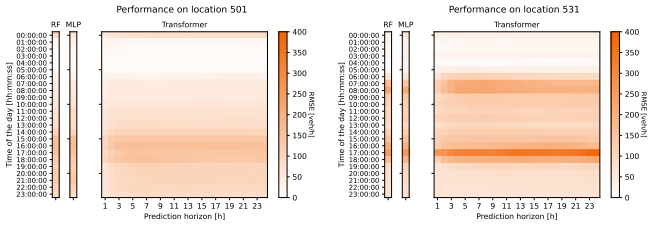
The performance of the transformer is evaluated and compared to the baseline models on four aspects. First, the performance on the train, validation, and test set is investigated. Next, the performance at different times of the day and prediction horizons is examined. Subsequently, the focus is on the days of the year on which the prediction models encounter difficulties. At last, an estimate of the uncertainty of the predictions is provided by analyzing the distribution of the relative errors and an example of the final predictions is provided.

1) Performance on different data sets: In Fig. 7 the RMSE of the random forest, MLP, and transformer are shown for both locations on the train, validation, and test set. The baseline prediction models have a similar performance on the test set, which indicates that there is no clear preference for either of the two based on these characteristics. Moreover, on location 531 both models seem to underfit the data, which will be elaborated on shortly at the end of this section.



(a) RMSE location 501 (b) RMSE location 531

Fig. 7: Performance characteristics on the train, validation, and test set for the random forest, multilayer perceptron, and transformer on locations 501 and 531.



(a) Location 501 (b) Location 531

Fig. 8: RMSE on the test set for the random forest, multilayer perceptron, and transformer. Shown for the different times of the day and prediction horizons.

The transformer outperforms the baseline prediction models for location 501 on the test set. On the contrary, the transformer is outperformed by the baseline models on location 531. Notable is the difference in the performance of the transformers on the train, validation, and test set for both locations. This highlights that the transformer is overfitting on the training set, which is unexpected because the performance on the validation set does not indicate overfitting. This implies that the validation set is not representative of the test set. For which, two reasons are investigated.

First, 2019 might not be a proper representation of the years 2017 and 2018, causing a discrepancy between the data sets. This is supported by the performance of the MLP and random forest on location 501, which indicate an inferior performance on the test set. Therefore, to look into the discrepancy between years, the entire data set is shuffled, after which it is divided into the train, validation, and test set. The performance of the corresponding baseline models is similar on the validation and test set, which indicates that the difference found by the original baseline models could indeed be caused by a discrepancy between years. Secondly, the transformer might indirectly already be subject to the validation data through the historical traffic flow and multi-step predictions. Therefore, the possibility of implementing a different validation set in the transformer for location 501 is investigated. Every fifth week of 2017/2018 is taken as the validation set, such that the data, indirectly seen is reduced. The transformer has a similar performance on the train set. However, the performance on the validation set now clearly indicates that the transformer is overfitting. This highlights that randomly selecting the validation set is not representative of the test set and overfitting occurs.

Therefore, both explanations seem to influence the discrepancy between the performance on the different data sets. It is undesired to implement the shuffled data set in the transformer, because the test set will not be valid anymore, due to the second explanation. Finding a more suitable validation set and designing a new transformer accordingly is expected to improve the model performance and is recommended for future research.

2) Performance on different prediction horizons and time of day: The performance over the prediction horizons shows that for location 501, the transformer outperforms the baseline models on all prediction horizons but is especially superior on the first 6 prediction hours. For location 531, the transformer is advantageous, up to a horizon of 3 hours. The traffic flow depends on the time of the day. Therefore, additionally, the RMSE corresponding to each time and prediction horizon on the test set is calculated for both locations and shown in Fig. 8. By looking into the performance of both transformers at different times of the day in combination with the deviation in traffic flow noticed at that hour, a few things are noticed.

When subject to a small range of traffic flows, the performance of the multiple models is similar and additionally remains constant over the horizons for the transformer. However, when subject to a broad range of traffic flows, the performance increases with a decrease in the horizon and the transformer outperforms the baseline prediction models on at least the first few horizons. This is reasonable because when subject to a small range of traffic flows, the traffic flow is mainly based on the time of the day and maybe the dow. Therefore, by implementing the transformer or decreasing the horizon, the prediction will only slightly be influenced. On the other hand, when subject to a broader range, the traffic flow is more complicated, which might also be shown in the previous traffic flow, highlighting the advantage of the transformer and a decrease in the horizon.

Location 531 mainly contains commuter traffic and consequently primarily shows the first type of behavior. The reason that for location 501 the transformer is superior for a relatively long prediction horizon is that the location has many subsequent hours of the day with a broader range of traffic flow values. Consequently, irregular behavior might be indicated earlier. This indicates that the transformer method is beneficial on both short and long prediction horizons when the locations are subject to irregular traffic flow.

3) Performance throughout the year: To investigate when the prediction models encounter difficulties and whether an underlying cause can be found the performance throughout the year is investigated.

The clustering analyses for location 501 showed a clear difference in traffic behavior during summer and winter. The baseline models encounter difficulties during summer, whereas the transformer behavior remains similar over the year. More-

over, two clusters contained irregular days. Even though the transformer outperforms the baseline models on these days, the transformer still has relatively high errors. Therefore, more research should be done to investigate additional important features, which might cause this behavior.

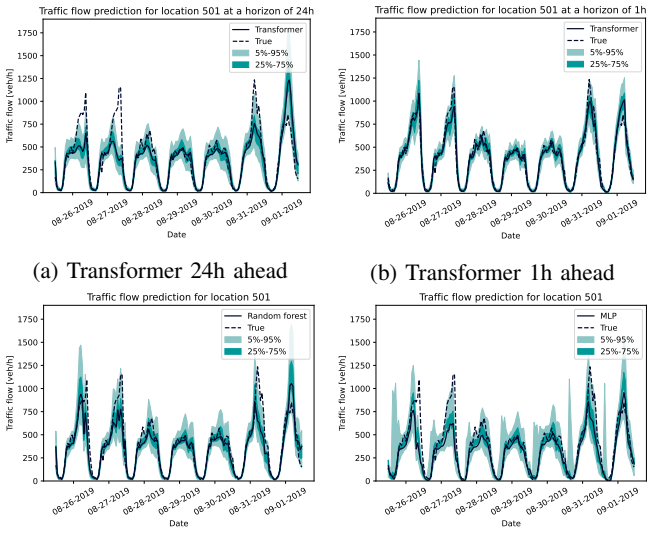
For location 531, the overall performance is similar. In the cluster analysis, the effect of school vacations was clearly shown, which all models can anticipate. Notable is a poor performance in March. The clusters show that most days in March 2017 are assigned to a separate cluster. The corresponding traffic flow is found to contain implausible traffic flow. The baseline prediction models are unable to model this behavior and consequently encounter large errors in the train and validation set. This explains why the baseline prediction models have a better performance on the test and is found to cause the high RMSE shown in Fig. 8b. Whether this traffic flow data is invalid and should be removed, or is caused by an external feature that is not implemented yet, such as construction works, is unknown. Therefore, this data should not just be removed, but for future research, the cause behind this behavior should be investigated.

4) Uncertainty of the predictions: The last step is to provide an estimate of the uncertainty of the predictions made, which is based on the performance on the test set. The uncertainty of the predictions is investigated by first calculating the relative error $e_{t,h}$ at start time step t and prediction horizon h. Next, the distribution of the errors is investigated to decide how these boundaries should be derived.

Due to the numerous positive outliers in the relative errors, caused by implausible and unpredictable low traffic flow, it is chosen to derive the uncertainty ranges by percentiles of the relative errors. The investigated percentiles (p) are 5%, 25%, 50%, 75%, and 95%, such that a boundary $\gamma_{t,h,p}$, for each p is obtained. These boundaries can be used to indicate the uncertainty ranges. This highlights that the MLP has a larger range in uncertainty values than the random forest, which implies a small preference towards the random forest, regarding the two baseline models. Moreover, the lowest uncertainty ranges are obtained by the transformer and are shown to decrease with a decrease in the prediction horizon.

5) Final prediction of the transformer: During inference, it is beneficial to apply the transformer at each hour of the day and make a multistep prediction for the next 24 hours. Next, the predictions can be updated every hour because the performance increases with a decrease in the horizon. To embody the final predictions, the results corresponding to the last week of August 2019 and the first week of January 2019 at location 501 are given. These correspond to a relatively difficult week and a regular week, respectively. The predictions made by the transformer at a horizon of 24 hours and 1 hour and by the baseline models are provided in Fig. 9 and Fig. 10.

These illustrate that at first, the transformer is not able to predict irregular behavior. However, when the prediction horizon is decreased, the performance increases significantly and the transformer can predict the irregular traffic flow. Moreover, the uncertainty ranges decrease as well. The random forest



(a) Transformer 24h ahead

(b) Transformer 1h ahead

(c) Random forest

(d) Multilayer perceptron

Fig. 9: Traffic flow predictions for the first week of August 2019, made by the transformer at a prediction horizon of 24 and 1 hours, the random forest, and multilayer perceptron.

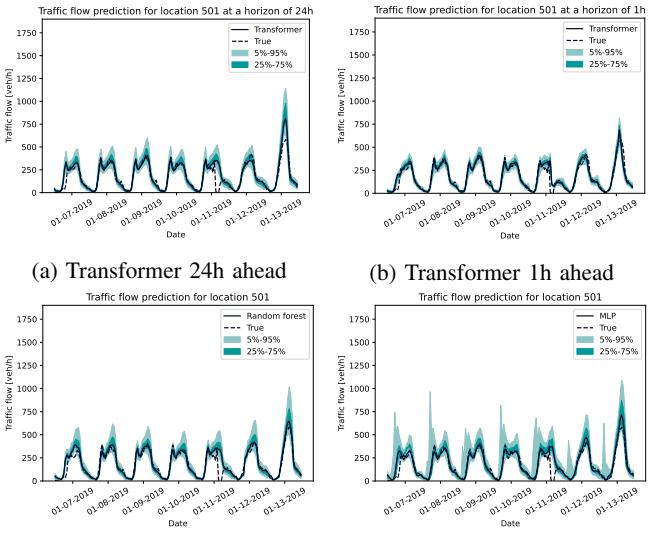
anticipates irregular traffic behavior and has a performance comparable to the transformer at a prediction horizon of 24 hours. This is in contrast to the MLP, which shows a worse performance. The uncertainty ranges of the baseline models are relatively large. In addition, significant peaks in the $5\% - 95\%$ range of the MLP occur. These are caused by large relative errors when subject to little traffic, causing high peaks when a bit of traffic flow is predicted at these hours.

The traffic flow predictions during the first week of January 2019 are shown in Figure 10. These figures highlight that all prediction models can predict the traffic flow well. Again, the performance of the transformer improves with the decrease in the prediction horizon, and the MLP shows high peaks in the uncertainty ranges.

The ability of the transformer to model irregular traffic flow behavior is shown. In addition, the applicability of the transformer on long prediction horizons is also indicated. Analyses have shown that for location 501 the transformer outperforms the baseline models on all prediction horizons and is especially superior in the first six hours. On the other hand, the transformer for location 531 is only superior up to the first three prediction horizons, which suggests that the adequacy of the transformer is location-dependent. Moreover, the transformer has the smallest uncertainty ranges, which are additionally shown to decrease with a decrease in the horizon.

VII. Conclusions and Recommendations

In this paper, a generic transformer-based prediction model is presented and the adequacy of external features and the transformer on long-term traffic flow predictions are highlighted. Moreover, the adequacy of the transformer is found to be location-dependent and thought to be promising for



(a) Transformer 24h ahead

(b) Transformer 1h ahead

(c) Random forest

(d) Multilayer perceptron

Fig. 10: Traffic flow predictions for the first week of January 2019, made by the transformer at a prediction horizon of 24 and 1 hour, the random forest, and the multilayer perceptron

locations that are subject to irregular traffic on both short and long horizons.

The objective was the obtain insights into traffic behavior to provide multi-modal itineraries and be able to influence traffic flows. The predictions 24 hours ahead already provide a good indication of the traffic behavior, which allows providing itinerary recommendations. Moreover, because most itineraries take less than a few hours, these can be updated before departure. On these accounts, the transformer is thought to be promising for these applications.

Nevertheless, multiple subjects lend themselves to further research. Most importantly, a new validation set should be sought, such that overfitting of the transformer is avoided. Subsequently, further extensions can be considered to expand the transformer. First, additional correlation analyses should be performed to investigate the influence of additional external features, such as events or construction works. Secondly, computational constraints limited the number of past input features. However, more useful data can be included without increasing the total model complexity by concatenating recent, daily, and weekly data, as done in [3, 13, 15]. Moreover, the network can be extended by including spatial features, which allows for correlations between different locations and is shown to be advantageous in traffic flow predictions [6, 24, 42].

Finally, to further investigate the genericity, the prediction models should be applied to additional locations based on different characteristics and if necessary, the feature set should be extended accordingly.

Acknowledgment

I would like to thank Alexander Koek for his support and his profound ability to look at challenges from a fresh perspective. I also wish to thank Bart De Schutter for his constructive feedback.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 24, 2011.

[3] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu. Traffic transformer: capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS*, 24(3):736–755, 2020.

[4] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078.*, 2014.

[5] F. Chollet and others. Keras. *Retrieved from https://github.com/fchollet/keras*, 2015.

[6] Z. Cui, K. Henrickson, R. Ke, Z. Pu, and Y. Wang. Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894, 2020.

[7] A. Downs. Traffic: why it's getting worse, what government can do. *(No. Policy Brief# 128) Washington, DC: Brookings Institution*, 2004.

[8] B. Du, H. Peng, S. Wang, M. Z. A. Bhuiyan, L. Wang, Q. Gong, L. Liu, and J. Li. Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):972–985, 2020.

[9] W. Duan, L. Jiang, N. Wang, and H. Rao. Pre-trained bidirectional temporal representation for crowd flows prediction in regular region. *IEEE Access*, 7:143855–143865, 2019.

[10] S. Fang, X. Pan, S. Xiang, and C. Pan. Meta-MSNet: meta-learning based multi-source data fusion for traffic flow prediction. *IEEE Signal Processing Letters*, 28:6–10, 2021.

[11] R. Fu, Z. Zhang, and L. Li. Using LSTM and GRU neural network methods for traffic flow prediction. In *In 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 324–328. IEEE, 2016.

[12] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016.

[13] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):922–929, 2019.

[14] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan. Deep spatial-temporal 3D convolutional neural networks for

traffic data forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3913–3926, 2019.

[15] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. *A Field Guide to Dynamical Recurrent Neural Networks, IEEE Press*, 2001.

[18] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] A. Koesdwiady, R. Soua, and F. Karray. Improving traffic flow prediction with weather information in connected cars: a deep learning approach. *IEEE Transactions on Vehicular Technology*, 65(12):9508–9517, 2016.

[20] L. Li, L. Qin, X. Qu, J. Zhang, Y. Wang, and B. Ran. Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm. *Knowledge-Based Systems*, 172:1–14, 2019.

[21] R. Li, Y. Hu, and Q. Liang. T2F-LSTM method for long-term traffic volume prediction. *IEEE Transactions on Fuzzy Systems*, 28(12):3256–3264, 2020.

[22] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y. Wang, and X. Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems*, 32:5243–5253, 2019.

[23] Y. Li, S. Chai, Z. Ma, and G. Wang. A hybrid deep learning framework for long-term traffic flow prediction. *IEEE Access*, 9:11264–11271, 2021.

[24] Y. Li, R. Yu, C. Shahabi, and Y. Liu. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2018.

[25] Ziyue Li, Hao Yan, Chen Zhang, and Fugee Tsung. Long-short term spatiotemporal tensor prediction for passenger flow profile. *IEEE Robotics and Automation Letters*, 5(4):5010–5017, 2020.

[26] B. Lim, S. Arık, N. Loeff, and T. Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

[27] O. Mohammed and J. Kianfar. A machine learning approach to short-term traffic flow prediction: a case study of interstate 64 in Missouri. *IEEE International Smart Cities Conference (ISC2)*, pages 1–7, 2018.

[28] K. Pearson. Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[30] D. Roberts and S. F. Brown. Identifying calendar-correlated day-ahead price profile clusters for enhanced energy storage scheduling. *Energy Reports*, 6:35–42, 2020.

[31] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson. Structured sequence modeling with graph convolutional recurrent networks. *In International Conference on Neural Information Processing*, Springer, Cham:362–373, 2016.

[32] C. Spearman. The proof and measurement of association between two things. 1961.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *In Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[34] Z. Wang, X. Su, and Z. Ding. Long-term traffic prediction based on LSTM encoder-decoder architecture. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2020.

[35] J.H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[36] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang. Graph WaveNet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

[37] M. Xu, W. Dai, C. Liu, X. Gao, W. Lin, G.J. Qi, and H. Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

[38] S. Yang and S. Qian. Understanding and predicting travel time with spatio-temporal features of network traffic flow, weather, and incidents. *IEEE Intelligent Transportation Systems Magazine*, 11(3):12–28, 2019.

[39] B. Yu, H. Yin, and Z. Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2018.

[40] D. Zhang and M. R. Kabuka. Combining weather condition data to predict traffic flow: a GRU based deep learning approach. *IET Intelligent Transport Systems*, 12(7):578–585, 2017.

[41] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D. Y. Yeung. GaAN: Gated attention networks for learning on large and spatiotemporal graphs. *arXiv preprint arXiv:1803.07294*, 2018.

[42] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2018.

Bibliography

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 24, 2011.

[3] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[4] J. Bergstra, D. Yamins, and D. D. Cox. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. *International Conference on Machine Learning*, 28:115–123, 2013.

[5] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu. Traffic transformer: capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS*, 24(3):736–755, 2020.

[6] Central Bureau for Statistics. Motorvoertuigen; voertuigtype; postcode en regio's, 1 januari. *Retrieved from: https://opendata.cbs.nl/statline/#/CBS/nl/dataset/37209hvv/table?ts=1620310075859*, 2020.

[7] Central Bureau for Statistics. Verkeersprestaties motorvoertuigen; kilometers, voertuigsoort, grondgebied. *Retrieved from https://opendata.cbs.nl/statline/#/CBS/nl/dataset/80302ned/table?dl=2BBD0*, 2020.

[8] H. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah. Wide & deep learning for recommender systems. *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10, 2016.

[9] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078.*, 2014.

[10] F. Chollet and others. Keras. *Retrieved from https://github.com/fchollet/keras*, 2015.

[11] Z. Cui, K. Henrickson, R. Ke, Z. Pu, and Y. Wang. Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894, 2020.

[12] C. Ding, D. Wang, X. Ma, and H. Li. Predicting short-term subway ridership and prioritizing its influential factors using gradient boosting decision trees. *Sustainability (Switzerland)*, 8(11), 2016.

[13] A. Downs. Traffic: why it's getting worse, what government can do. *(No. Policy Brief# 128) Washington, DC: Brookings Institution*, 2004.

[14] B. Du, H. Peng, S. Wang, M. Z. A. Bhuiyan, L. Wang, Q. Gong, L. Liu, and J. Li. Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):972–985, 2020.

[15] W. Duan, L. Jiang, N. Wang, and H. Rao. Pre-trained bidirectional temporal representation for crowd flows prediction in regular region. *IEEE Access*, 7:143855–143865, 2019.

[16] S. Fang, X. Pan, S. Xiang, and C. Pan. Meta-MSNet: meta-learning based multi-source data fusion for traffic flow prediction. *IEEE Signal Processing Letters*, 28:6–10, 2021.

[17] P.I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[18] R. Fu, Z. Zhang, and L. Li. Using LSTM and GRU neural network methods for traffic flow prediction. In *In 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 324–328. IEEE, 2016.

[19] Gemeente Amsterdam. Scale up | Bezoekersstromen. *https://innovatiepartners.nl/project/scale-up-or-bezoekersstromen#top*, 2020.

[20] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016.

[21] J. Grigsby, Z. Wang, and Y. Qi. Long-range transformers for dynamic spatiotemporal forecasting. *arXiv preprint arXiv:2109.12218*, 2021.

[22] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):922–929, 2019.

[23] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan. Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3913–3926, 2019.

[24] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] D. Hillen. Monitoring ten behoeve van reistijdinformatie: catalogus van monitoringsystemen gericht op reistijdinformatie. *Catalogus van monitoringsystemen gericht op reistijdinformatie*, 2006.

[27] C. Q. Ho, D. A. Hensher, C. Mulley, and Y. Z. Wong. Potential uptake and willingness-to-pay for Mobility as a Service (MaaS): a stated choice study. *Transportation Research Part A: Policy and Practice*, 117:302–318, 2018.

[28] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. *A Field Guide to Dynamical Recurrent Neural Networks, IEEE Press*, 2001.

[29] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. *International Conference on Learning and Intelligent Optimization*, pages 507–525, 2011.

[30] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] A. Koesdwiady, R. Soua, and F. Karray. Improving traffic flow prediction with weather information in connected cars: a deep learning approach. *IEEE Transactions on Vehicular Technology*, 65(12):9508–9517, 2016.

[32] L. Li, L. Qin, X. Qu, J. Zhang, Y. Wang, and B. Ran. Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm. *Knowledge-Based Systems*, 172:1–14, 2019.

[33] R. Li, Y. Hu, and Q. Liang. T2F-LSTM method for long-term traffic volume prediction. *IEEE Transactions on Fuzzy Systems*, 28(12):3256–3264, 2020.

[34] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y. Wang, and X. Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems*, 32:5243–5253, 2019.

[35] Y. Li, S. Chai, Z. Ma, and G. Wang. A hybrid deep learning framework for long-term traffic flow prediction. *IEEE Access*, 9:11264–11271, 2021.

[36] Y. Li, R. Yu, C. Shahabi, and Y. Liu. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2018.

[37] B. Lim, S. Arık, N. Loeff, and T. Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

[38] D. Lim, M. Lee, and J. Seok. Long term traffic prediction in highway using parallel CNN. *In 2020 IEEE 5th International Conference on Intelligent Transportation Engineering (ICITE)*, pages 107–110, 2020.

[39] O. Mohammed and J. Kianfar. A machine learning approach to short-term traffic flow prediction: a case study of interstate 64 in Missouri. *IEEE International Smart Cities Conference (ISC2)*, pages 1–7, 2018.

[40] A. Moussavi-Khalkhali and M. Jamshidi. Leveraging machine learning algorithms to perform online and offline highway traffic flow predictions. *Proceedings - 2014 13th International Conference on Machine Learning and Applications*, pages 419–423, 2014.

[41] NDW. NDW Interface beschrijving Actuele verkeersgegevens. 2010.

[42] B. Osborne and R. Bellis. The congestion con: how more lanes and more money equals more traffic. *Washington, DC: Transportation for America*, 3(21), 2020.

[43] C. M. Own, F. Sha, and W. Tao. Triplet decoders neural network ensemble system and t-conversion for traffic speed sequence prediction. *IEEE Access*, 7:162070–162082, 2019.

[44] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review, 2019.

[45] K. Pearson. Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895.

[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[47] W. Polman. Voertuigdetectie: wensen en mogelijkheden. *commissioned by: ministerie van Verkeer en Waterstaat/Rijkswaterstaat Adviesdienst Verkeer en Vervoer*, 2002.

[48] Provincie Flevoland and Ministerie van Economische Zaken en Klimaat. SBIR oproep Smart Mobility Flevoland. *https://www.rvo.nl/subsidie-en-financieringswijzer/sbir/sbir-oproep-smart-mobility-flevoland*, 2021.

[49] D. Roberts and S. F. Brown. Identifying calendar-correlated day-ahead price profile clusters for enhanced energy storage scheduling. *Energy Reports*, 6:35–42, 2020.

[50] D. E. Rumelhart, G. E. Hint, and R. J. Williams. Learning internal representations by error propagation. *Learning internal representations by error propagation*, (California Univ San Diego La Jolla Inst for Cognitive Science), 1985.

[51] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson. Structured sequence modeling with graph convolutional recurrent networks. *In International Conference on Neural Information Processing*, Springer, Cham:362–373, 2016.

[52] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25, 2012.

[53] C. Spearman. The proof and measurement of association between two things. 1961.

[54] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[55] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. Choudhury, and A. K. Qin. A survey on modern deep neural network for traffic prediction: trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020.

[56] F. Toqué, M. Khouadjia, E. Come, M. Trepanier, and L. Oukhellou. Short & long term forecasting of multimodal transport passenger flows with machine learning methods. *In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE:560–566, 2017.

[57] S. M. Turner. Guidelines for developing ITS data archiving systems. *No. Report 2127-3*, 2001.

[58] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *In Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[59] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[60] Z. Wang, X. Su, and Z. Ding. Long-term traffic prediction based on LSTM encoder-decoder architecture. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2020.

[61] J.H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[62] W. Weijermars. Analysis of urban traffic patterns using clustering. *Netherlands TRAIL Research School*, 41, 2007.

[63] W. Weijermars and E. van Berkum. Analyzing highway flow patterns using cluster analysis. *In Proceedings. 2005 IEEE Intelligent Transportation Systems*, pages 308–313, 2005.

[64] A. Wilson and R. de Groot. *Handboek verkeerslichtenregelingen*. CROW te Ede, 2014.

[65] Z. Wu, S. Pan, F. Chen, Long G., C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2019.

[66] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang. Graph WaveNet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

[67] M. Xu, W. Dai, C. Liu, X. Gao, W. Lin, G.J. Qi, and H. Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

[68] S. Yang and S. Qian. Understanding and predicting travel time with spatio-temporal features of network traffic flow, weather, and incidents. *IEEE Intelligent Transportation Systems Magazine*, 11(3):12–28, 2019.

[69] B. Yu, M. Li, J. Zhang, and Z. Zhu. 3D graph convolutional networks with temporal graphs: A spatial information free framework for traffic forecasting. *arXiv preprint arXiv:1903.00919*, 2019.

[70] B. Yu, H. Yin, and Z. Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2018.

[71] T. Yu and H. Zhu. Hyper-parameter optimization: a review of algorithms and applications. *arXiv preprint arXiv:2003.05689*, 2020.

[72] D. Zang, J. Ling, Z. Wei, K. Tang, and J. Cheng. Long-term traffic speed prediction based on multiscale spatio-temportal feature learning network. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3700–3709, 2019.

[73] D. Zhang and M. R. Kabuka. Combining weather condition data to predict traffic flow: a GRU based deep learning approach. *IET Intelligent Transport Systems*, 12(7):578–585, 2017.

[74] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D. Y. Yeung. GaAN: Gated attention networks for learning on large and spatiotemporal graphs. *arXiv preprint arXiv:1803.07294*, 2018.

[75] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi. DNN-based prediction model for spatio-temporal data. *In Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 1–4, 2016.

[76] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2018.

[77] Z. Zheng, Y. Yang, J. Liu, H. N. Dai, and Y. Zhang. Deep and embedded learning approach for traffic flow prediction in urban informatics. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3927–3939, 2019.

Glossary

List of Acronyms

dow	day of the week
KNMI	Koninklijk Nederlands Meteorologisch Instituut
MAE	mean absolute error
MLP	multilayer perceptron
NDW	Nationaal Dataportaal Wegverkeer
ReLU	Rectified Linear Unit
RMSE	root mean squared error