Project

AI in Transportation

Author: Johanna Schaefer

17.10.2025

abstract

Contents

1. Introduction

In modern transport systems, continuous collection of traffic data is essential for providing real-time information on speed and traffic flow. Sensors on road portals provide important data, but if individual sensors fail or provide incomplete data, this can affect the accuracy of predictions. The central question of this work is therefore: How well can missing sensors be compensated for by data from other sensors in the same portal or from neighbouring portals? To answer this question, a regression approach is used in which the measured values of a target sensor are predicted on the basis of neighbouring sensors. The study examines whether sensors within the same portal provide better prediction accuracy than sensors from a neighbouring portal.

2. Descriptive analysis

To begin the project of creating predictive models of the data provided, the data needs to be analysed.

2.1. Data description

The data set used for this project consists of speed and traffic flow data from several sensors on the motorway near Stockholm. The data comes from 29 sensors belonging to 8 different portals on a section heading south. Every sensor measures the speed and flow in one line of the motorway. There are inflows and outflows within this section between the portals as can be seen in figure 2.1.

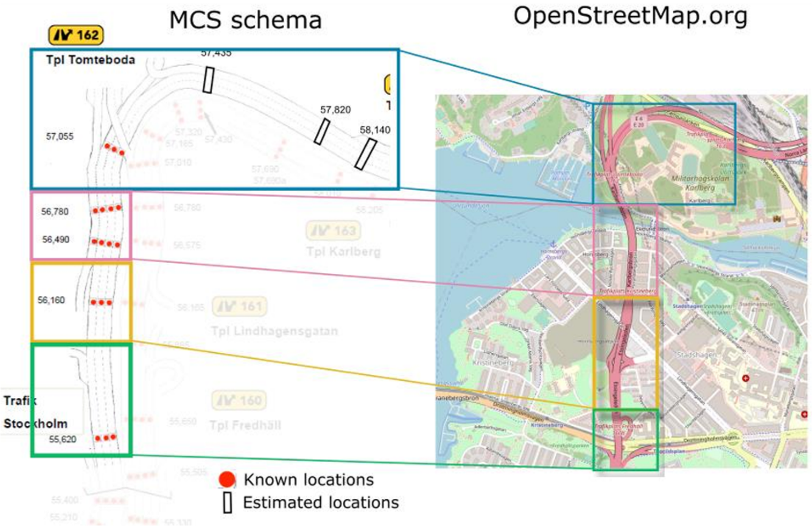


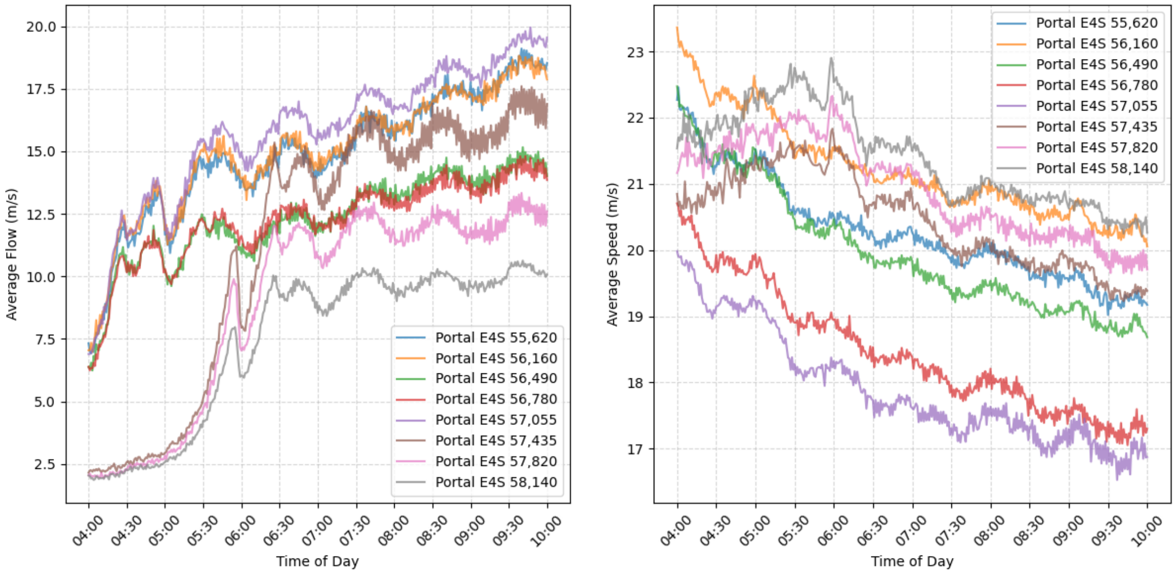
Figure 2.1: Overview motorway section and portals

Table B.1 shows the portals and the correspondings sensors.

Over a total of 214 days, speed and flow data were recorder between 4 AM and 10 AM with a temporal resolution of one minute. Speed is measured in $\mathrm{m/s}$, while flow is quantified as the number of vehicles per minute.

2.2. Speed and Flow per portal

Between 4 AM and 10 AM, the flow of vehicles shows a generally increasing trend (see figure B.1), indicating a continued build up of traffic volume as the morning progresses. The rate at which it in increasing is slowing down when approaching the 10 AM. In contrast, the average speed tends to decline over the same period, reflecting growing congestion. This inverse relationship between flow and speed is characteristic of peak-hour dynamics, where higher vehicle density leads to reduced travel speeds. Furthermore, it can be seen that it seems like there are two different groups of portals. Portal 55620, 56160, 56490, 56780 and 57055 are more similar to each other than to the other three, that also seem to form a group.



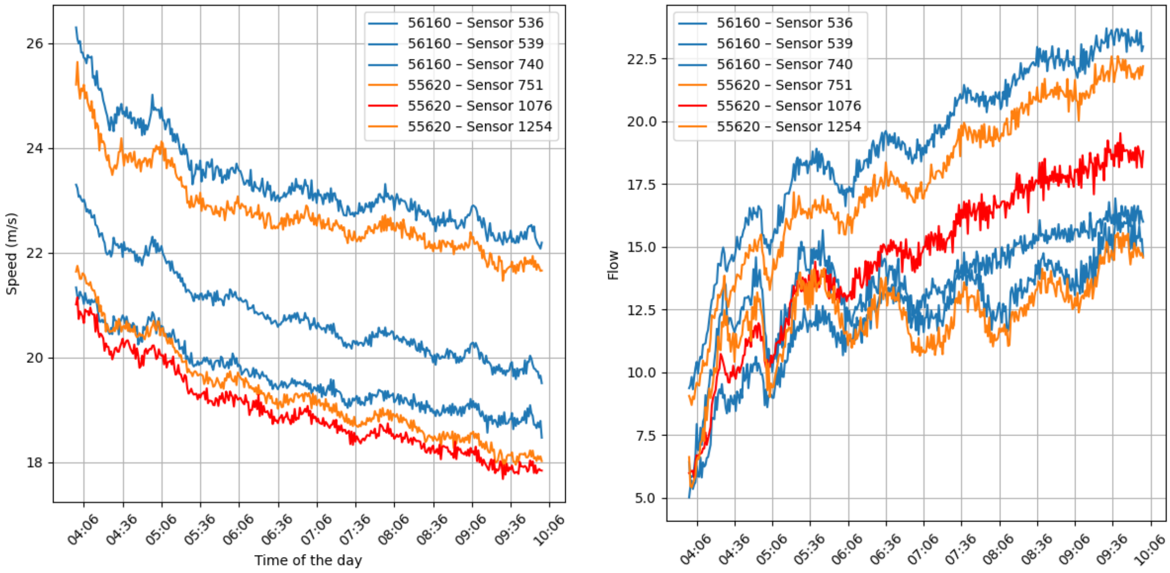
(a) Flow

(b) Speed

Figure 2.2: Flow and Speed over the day in the different portals

2.3. Speed and Flow per sensor in portal 55620 and 56160

In the previous chapter the speed and flow was averaged over the portal. Furthermore, it should also be looked into the daily profil for different sensors.



(a) Flow

(b) Speed

Figure 2.3: Flow and Speed for the sensors in portal 55620 and 56160

2.4. Correlation

Furthermore, the correlation between different sensors was analysed for both speed and flow. It is important to note that no time lag was considered in this initial assessment. In addition to variations in speed and flow across sensors, spatial separation plays a significant role: the greater the distance between sensors, the more pronounced the temporal offset and divergence in observed traffic patterns. In figure 2.4 the sensors are displayed grouped by the portals they are part of. The thicker black lines indicate the portals. Also in this correlation matrix, the two different groups of portals can be identified. Furthermore, speed and flow show different patterns. It can be seen that for the speed the correlation is generally higher within the same portal (indicated by high density of red squares close to the diagonal). On the other side, this is not necessarily the case for the flow values. It can be seen that the red squares are more spread out, showing that the correlation is also partially quite high with sensor data from other portals.

2.5. Clustering as lane identification

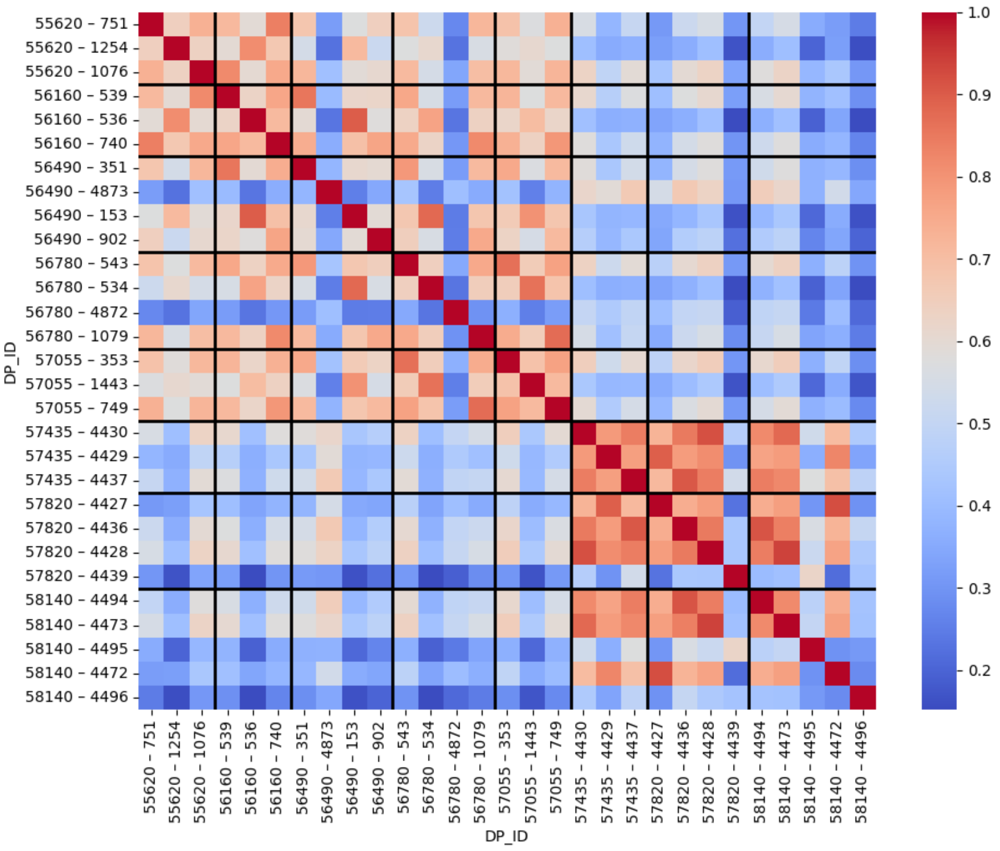
As part of the data analysis, lane identification was performed using unsupervised clustering. The underlying assumption is that different lanes exhibit distinct traffic characteristics in terms of speed and flow. Typically, faster lanes are located on the left (e.g., overtaking lanes), while slower lanes are positioned on the right (e.g., exit or merging lanes). It was hoped to find these behaviour patterns in the sensor data to infer lane structure. To uncover these latent lane groupings, K-Means clustering was used. It was chosen among several other clustering methods as it is the easiest method to control the number of clusters which should in the best case- represent the lanes. Based on the pattern that was observed in the correlation matrix showing two different portals groups, they were clustered separately.

For the five portals, the number of clusters was set to 4 according to the number of lanes. The clustering was very successful in the sense that each portal has only one sensor per cluster as can be seen in table 2.1, which suggests that the cluster lane identification approach is appropriate.

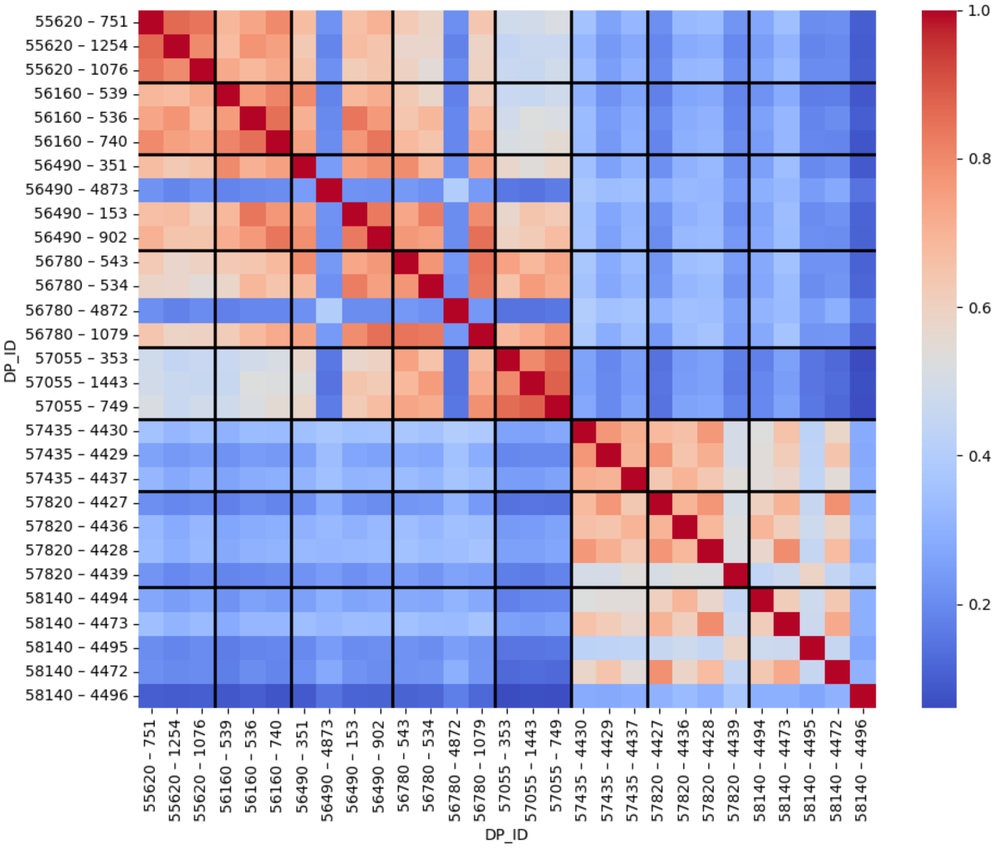
Table 2.1: The number and name of the sensor that belong to every cluster for every portal and the most likely lane that cluster corresponds to

Portal/Cluster	0 (exit/marging)	1(left)	2(middle)	3 (right)
55620	1 (1076)	0	1(1254)	1(751)
56160	1(539)	0	1(536)	1(740)
56490	1(351)	1(4873)	1(153)	1(902)
56780	1(543)	1(4872)	1(534)	1(1079)
57055	1(353)	0	1(1443)	1(749)

Figure 2.5 shows the typical day profil for the different clusters that were identified.

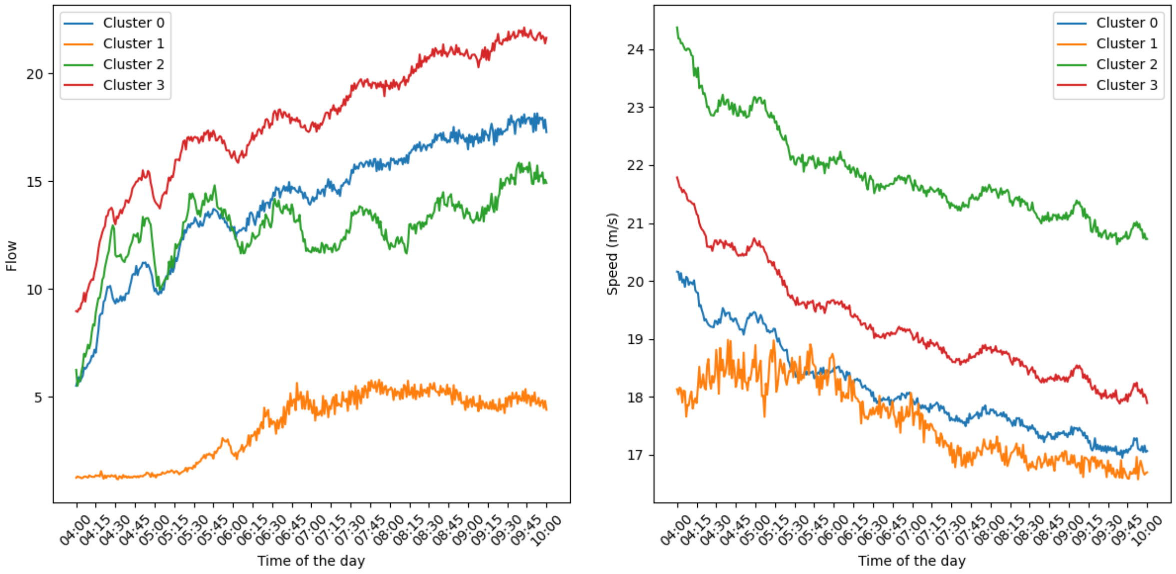


(a) Flow



(b) Speed

Figure 2.4: Correlation between speed and flow in different sensors



(a) Flow

(b) Speed

Figure 2.5: Flow and Speed profile of the 4 distinct clusters

Based on these patterns it is very likely -though not proven - that cluster 2 corresponds to the leftest lane showing the highest speed, cluster 3 corresponds to the middle lane with a lower speed, cluster 0 the the right lane and cluster 1 to the lane where cars enter and leave the motorway with the lowest flow. The clustering for the three other sensors can be found in the appendix.

3. Problem formulation

The primary objective of this project is to evaluate how well missing sensors can be compensated by using data from other sensors.
In real-life traffic management system rely heavily on the data from sensors. However, sensors may fail due to maintenance or technical failure. Thus, it becomes crucial to predict data for the missing sensors using data from nearby sensors.
In this study, the prediction problem is formulated as a regression task, where the target variable represents either the average speed or the summed flow over a future 15-minute interval. The models are trained to predict this value based on lagged features from other sensors.
Concretely, the analysis focuses on one reference sensor (Sensor 1076 in Portal 55620). Its values are predicted using two distinct data sources: (1) the two other sensors within the same portal (Sensors 751 and 1254), and (2) the three sensors from the neighboring portal (Sensors 539, 536, and 740). Comparing the predictive performance between these two groups allows for assessing how effectively missing sensors can be replaced by spatially close sensors.

3.1. Hypothesis

The hypothesis of this study is that sensors located within the same portal provide better predictive power for a target sensor than sensors from a neighboring portal.

4. Methodology and evaluation methods

This section outlines the methodological framework used in this study, including data preprocessing, feature engineering, model development, and the evaluation metrics applied to assess model performance.

4.1. Data preprocessing

The raw dataset contained multiple issues that required preprocessing. The Datetime column was converted to a proper datetime format to allow time-based operations and sequence creation. The PORTAL column was cleaned and standardized $(\mathrm{PORTAL}_\mathrm{c}\mathrm{lean})$ to ensure consistent portal identifiers across sensors.
As for the question of this project only the data of portal 55620 and 5610 was needed, only this data was extracted for further processing.
As can be seen in figure 4.1 some speed and/or flow values are missing in the dataset for some sensors.

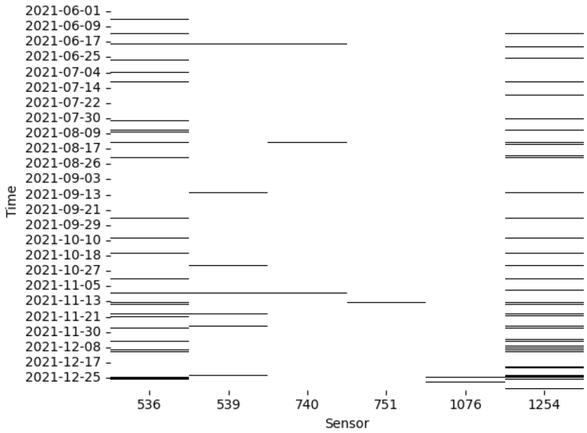


Figure 4.1: Missing values

Since models like Linear Regression cannot handle NaN values, these missing entries were imputed using forward and backward fill to ensure a complete dataset for modeling.

4.2. Feature engineering

To be able to predict the summed flow and average speed at the target sensor, the features had to be decided. Past values of speed and flow from the same and neighbouring sensors were used as lagged features. To justify the number of selected lagged observations,lag correlation plots were generated for both speed and flow of the target sensor with the lagged observation at the other sensors: Figure 4.2 shows that the speed shows a clear transition from a steep to a flat correlation curve at lag ≈ 15, with a secondary drop at lag ≈ 25. In the flow graph , on the other hand, no clear break is apparent. Considering the trade-off betweeen gaining more information and complexity , it was finally chosen to take 15 lagged features for both flow and speed.

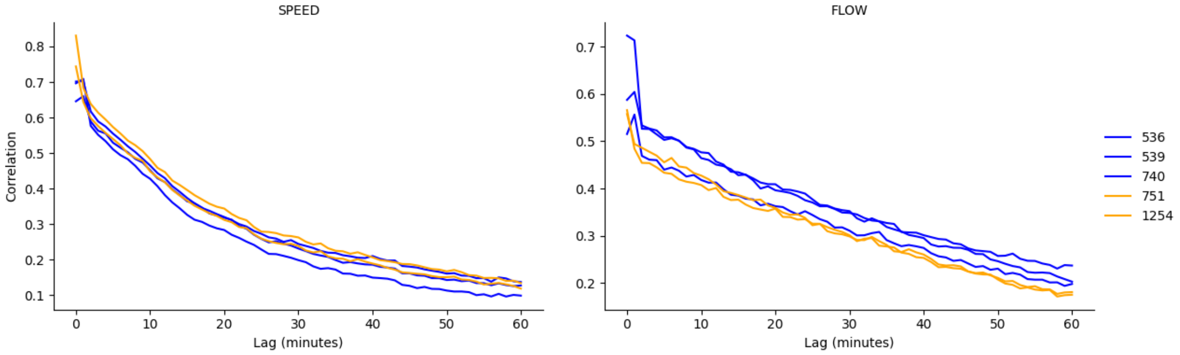


Figure 4.2: Correlation between the target value and the lagged valueso of the other sensors

After selecting the lagged features (past 15 observations for speed and flow), the dataset was transformed into a structured format suitable for model training. Each row of the new dataframe corresponds to a single time point and contains all lagged observations from the relevant sensors as input features (X), while the target variable (y) corresponds to the average speed or summed flow for the next 15-minute interval.

4.3. Model development

Several machine learning models were developed to predict the two target variables. The goal was to compare classical regression approaches with more advanced algorithms, including gradient boosting and neural networks, and to evaluate their performance under the same experimental setup. For the linear regression model, XG Boost and Feedforward Neural Network a random train-test split was applied, while a simple chronological split was used in the case of LSTM. The data was scaled with a Standard Scaler in the case of the two deep learning models.

4.3.1. Linear Regression Model

Linear regression was chosen as a baseline model due to its simplicity and interpretability. The model predicts the target variable (either average speed or summed flow) as a linear combination of the lagged features from the same or neighboring sensors.

4.3.2. XGBoost

XGBoost is a machine learning model that combines many small decision trees. Each tree learns from the mistakes of the previous trees, so that the predictions gradually improve, minimising the squared error as loss function [1]. It can also handle missing values better and is often more robust against outliers. To optimize the performance of the XGBoost model, a Randomized Search was performed to find the best combination of hyperparameters. The following parameters were considered and adjusted during the search:

- `n_estimators`: Number of boosting rounds (trees). A higher number of trees is equivalent to a higher training time and risk for overfitting
- `max_depth`:Maximum depth of a single tree, controlling model complexity, smaller trees reduce overfitting but can also lead to underfitting.
- `learning_rate`:Step size shrinkage used to prevent overfitting. Higher values speed up learning but increase the risk of overshooting minima, while smaller values require more trees for the same performance.

- `subsample`: Fraction of training samples used for each tree, lower values reduce overfitting, but increase variance„ potentially making predictions less stable.
- `colsample_bytree`:Fraction of features used for each tree. Less features reduce overfitting , but might omit important information if set too low.

The Randomized Search was performed on the training data only, using 3-fold cross-validation. The best model was selected based on the lowest root mean squared error (RMSE) from the cross-validation results.

4.3.3. NN forward model

This type of model consists of multiple fully connected layers.In an FNN, information flows in one direction only from the input layer through the hidden layers to the output layer. Each neuron applies a weighted sum of its inputs, adds a bias term, and passes the result through a nonlinear activation function. The weights are adjusted according to an optimiser, in this case the Adam optimiser. Before training, all input features were standardized using a StandardScaler to ensure that all variables contributed equally to the learning process and to improve convergence. Each model was trained for up to 100 epochs with a batch size of 32, using mean squared error (MSE) as the loss function and root mean squared error (RMSE) as the evaluation metric. Early stopping was applied with a patience of 5 epochs to prevent overfitting, while the learning rate was dynamically reduced when the validation error plateaued. In a Gridsearch the number of neurons per layer, the number of layers and the drop out rate were changed.

4.3.4. LSTM model

Lastly, an LSTM model was tried out. This is a special neural network that can process time-dependent data and was thus considered relevant for this project. Unlike simple networks, an LSTM can remember previous information and thus can often better capture temporal relationships.
In this project two different versions of an LSTM were tried out:

LSTM with 15 output values The model gives me 15 separate output values for the upcoming 15 minutes. After that, these 15 values are summed up in the case of the flow or averaged in the case of the speed.

LSTM with one aggregated output value The model gives one, already output which already corresponds to the aggregated value.

4.4. Model evaluation

To evaluate the performance of the regression models, three error metrics were applied: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2).

Mean Squared Error (MAE) The Mean Absolute Error measures the average magnitude of the prediction errors.A lower MAE indicates better performance.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{4.1}$$

Root Mean Squared Error (RMSE) RMSE, compared to MAE, emphasizes larger errors, which is useful when large prediction mistakes are particularly undesirable. A lower RMSE indicates better performance.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.2}$$

Coefficence of Determination R² The coefficient of determination (R²) describes how much of the variance in the target variable is explained by the model. An R^2 value close to 1 indicates that the model explains most of the variance, whereas values near 0 indicate weak explanatory power. R² allows for intuitive comparison also across speed and flow that have different units.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2} \tag{4.3}$$

5. Results and Analysis

This sections aims to show the performance from the four different models on the test data. Finally, the performance of the best model on the evaluation set is evaluated.

5.1. Linear Regression

As shown in Table 5.1, the linear regression model provides a baseline for both flow and speed prediction. For the flow the prediction is better using the neighbouring portal with better RMSE, MAE and r² scores, while for the speed the opposite is the case. Furthermore, the r² score indicates that the flow can in general be better predicted than the speed. However, overall performance remains limited due which can be seen in the rather low r²-score.

Table 5.1: Results obtained with linear regression on testset

	RMSE	MAE	R²
Flow - same portal	33.543	23.809	0.836
Flow - neighbour portal	28.474	19.559	0.882
Speed - same portal	0.861	0.462	0.709
Speed -neighbour portal	1.051	0.513	0.566

5.2. XGBoost

The XGBoost model (Table 5.2) significantly improves prediction performance compared to linear regression across all cases. This improvement can be attributed to XGBoost's ability to model non-linear relationships between features. Again, the neighbouring portal performs better in all metrics for the flow, and the same portal better for the speed.

Table 5.2: Results obtained with XGBoost on testset

	RMSE	MAE	R²
Flow - same portal	28.311	20.292	0.883
Flow - neighbour portal	24.096	16.871	0.915
Speed - same portal	0.812	0.401,	0.740
Speed -neighbour portal	0.930	0.420,	0.660

5.3. Feedforward Neural Network (FNN)

The feedforward neural network (Table5.3) achieves similar but slightly worse results than XGBoost suggesting that the relatively small dataset may limit the potential of deep learning models. The speed predicted from the same portal is the only except for this with a marginally higher r² and marginally lower RMSE-score. It also should be mentioned here that the training time is significantly higher than with the XGBoost model, making it less efficient for practical deployment despite its comparable performance.

Table 5.3: Results obtained with Neural Network on testset

	RMSE	MAE	R^2
Flow - same portal	28.720	20.481	0.879
Flow - neighbour portal	24.511	17.286	0.912
Speed - same portal	0.805	0.419	0.745
Speed -neighbour portal	0.934	0.437,	0.657

5.4. Long Short-Term Memory Network (LSTM)

The LSTM results (Tables 5.4 and 5.5) show more variation and overall slightly lower performance compared to the feedforward and XGBoost models. The difference between the 15-output and 1-output configurations shows that the way temporal prediction is structured also has a large impact on results showing a better performance when first 15 separate values are predicted and then aggregated, which can be due to smoothing loss. Although the LSTM is known to capture sequential patterns, its advantage over simpler models remains limited in this case. However, it should be noted that no extensive hyperparameter tuning was performed due to the already high training time. Furthermore, a simple chronological train-test split was used, as a random split was not appropriate given the temporal dependencies in the data, which likely increased the training difficulty.

Table 5.4: Results obtained with LSTM and 15 ouptputs on testset

	RMSE	MAE	R^2
Flow - same portal	40.725	25.909	0.782
Flow - neighbour portal	34.867	22.393	0.840
Speed - same portal	0.880	0.511	0.730
Speed -neighbour portal	1.038	0.560	0.624

Table 5.5: Results obtained with LSTM and 1 ouptputs on testset

	RMSE	MAE	R^2
Flow - same portal	0.541	0.328	0.772
Flow - neighbour portal	0.436	0.278	0.815
Speed - same portal	1.003	0.530	0.634
Speed -neighbour portal	1.085	0.576	0.572

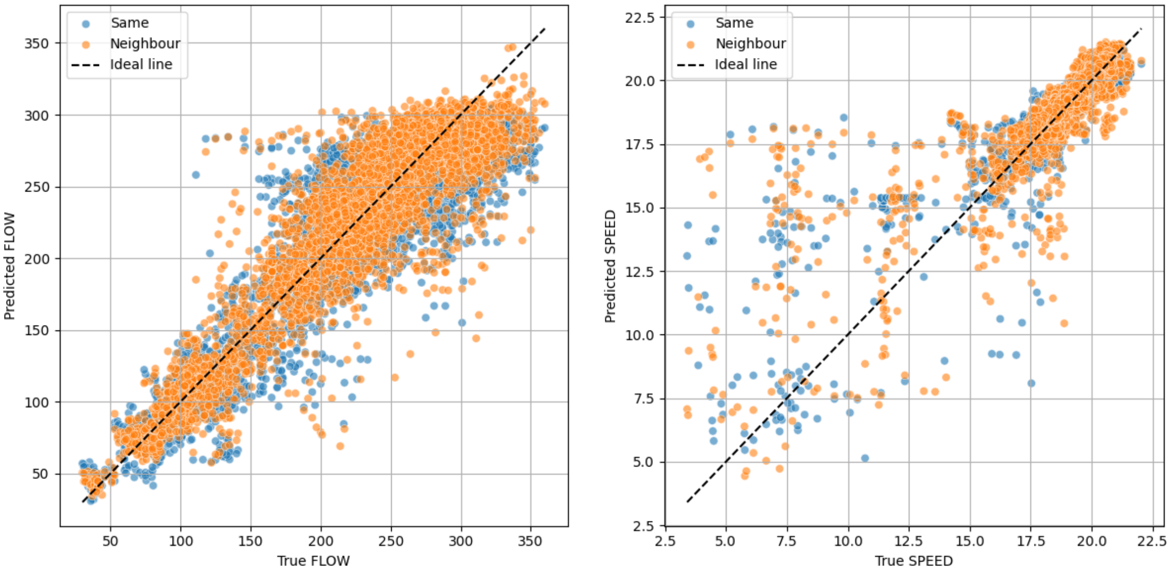
5.5. Evaluation on Final Evaluation Set

Based on the performance measured with the three different metrics, the XGBoost was chosen as final model. It did perform the best in all metrics for the flow predicted from the same and neighbouring portal and the speed predicted from the neighbouring portal Even though the result obtained from the FNN model for the speed predicted from the sensors in the same portal were slightly better, the XGBoost was still considered better when also taking the training

time into account.
On the evaluation set for the speed.

Table 5.6: Results obtained with XGBoost on the evaluation set

	RMSE	MAE	R²
Flow - same portal	31.127	23.841	0.801
Flow - neighbour portal	27.295	19.662	0.847
Speed - same portal	1.029	0.475	0.738
Speed -neighbour portal	1.173	0.533	0.659



(a) Flow **(b)** Speed

Figure 5.1

5.6. The influence of the distance on the model performance of the neighbouring portal

6. Model Deployment and Discussion

6.1. Model Robustness

In this project, the robustnesss of each model was evaluated by applying it to unseen test data. Furthermore, the XGBoost model was also evaluated on a separate evaluation set. A drop from the performance on the test set from the train-test split to the performance on the evaluation set could be seen, however was relatively small. The results suggest that the learned relationships between the target sensor and the neighbouring sensors are stable and can be applied to data outside the training period, demonstrating robustness in practical scenarios.

6.2. Handling Missing Data

As explained in 4.1 the models were trained with a filled data set. However a major advantage of XGBoost, which was chosen as final model is that it can handle Nan-Values, unlike other model types internally during prediction and training. This advantage was not used in this work upto now. To test this feature, the trained model was applied to the evaluation dataset containing missing values. The results show a notable decrease in performance, particularly for the flow prediction task. This can be attributed to the fact that the model had never been exposed to missing-value patterns during training. Additionally, the stronger drop for the flow can be explained by the fact that flow represents a sum and is therefore more sensitive to missing data. In contrast, the average speed is less affected, as it depends on relative rather than cumulative values.

Table 6.1: Results obtained with XGBoost on the evaluation set- with NaN

	RMSE	MAE	R²
Flow - same portal	44.891	29.646	0.554
Flow - neighbour portal	36.606	23.763	0.704
Speed - same portal	1.050	0.480	0.735
Speed -neighbour portal	1.186	0.536	0.662

6.3. Practical Deployment Considerations

In a real-world deployment, a separate model could be trained for each target sensor using historical data from its neighbouring sensors. In case of a total sensor failure, the pre-trained model could then be used to predict the missing values in real time, ensuring continuity of traffic monitoring. This approach would allow the traffic management system to operate reliably even when individual sensors are offline. Additionally, models could be periodically retrained to adapt to changing traffic conditions.

6.4. Limitations and Future Work

The main limitation of this work is that all analyses and model evaluations were based on data from a single lane and portal. Consequently, the conclusions cannot yet be generalized to other locations or traffic conditions. In future work, additional sensors and portals should be included to evaluate spatial transferability. A further limitation is that the XGBoost model was only trained with a full data set. Especially when looking at sensor data, missing values are a quite frequent phenomenon. The XGBoost should therefore be retrained with NAN-values to allow a better performance.

7. Summary of main findings and future works

This study investigated different machine learning approaches for short-term traffic prediction based on sensor data from a single highway portal. Four models — Linear Regression, XGBoost, a Feedforward Neural Network, and an LSTM network — were developed and compared using multiple performance metrics. Among these, XGBoost achieved the best overall results, combining strong predictive accuracy with robustness and computational efficiency.

Appendix

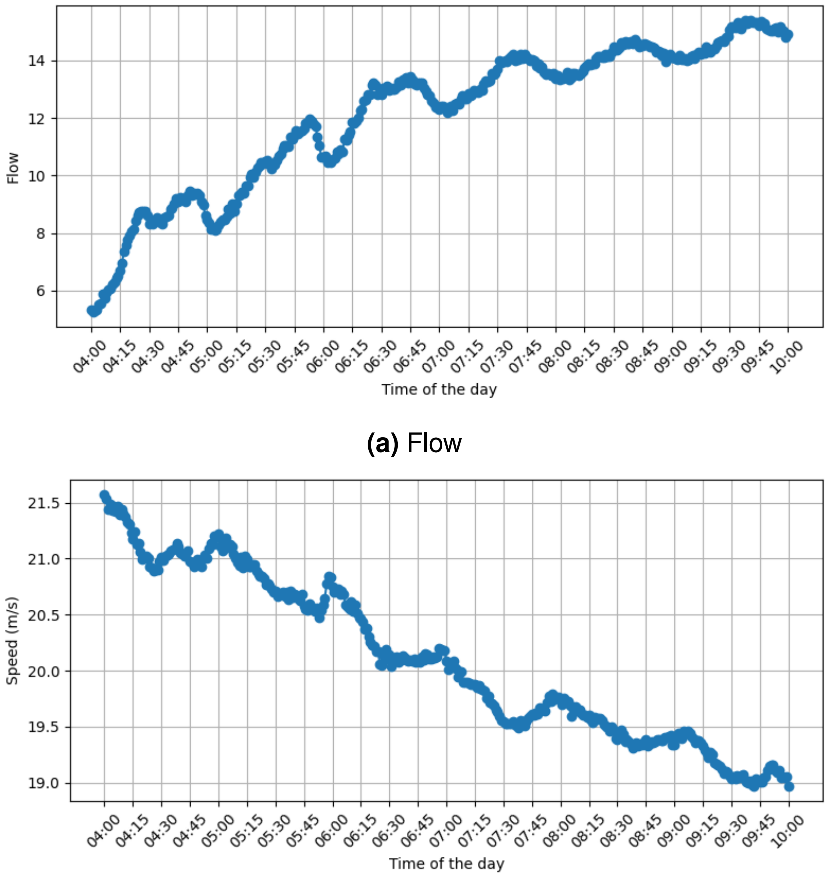
A. GitHub code link

The project can be accessed under: `https://github.com/Jojo18-20/Project_AI_Transportation`

B. Additional figures and tables

Table B.1: Portals and sensors

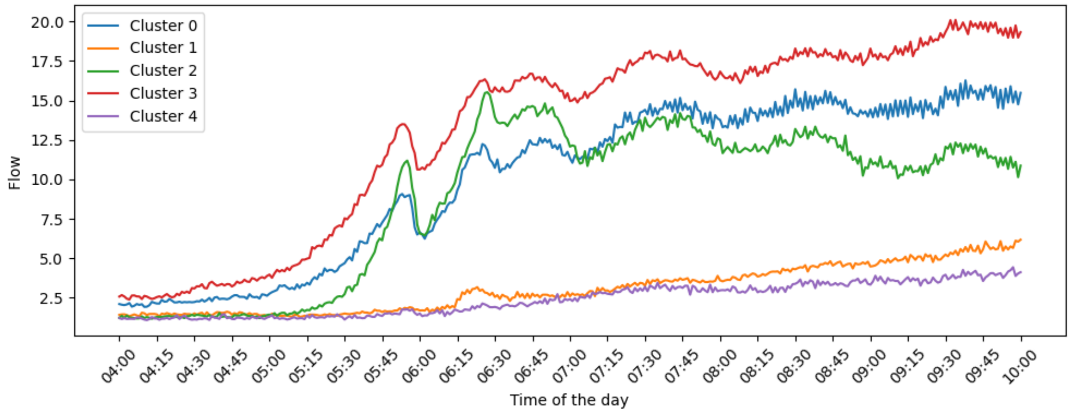
Portal	sensors
E4S 55620	751, 1254, 1076
E4S 56160	539, 536, 740
E4S 56490	351, 4873, 153, 902
E4S 56780	543, 534, 4872, 1079
E4S 57055	353, 1443, 749
E4S 57435	4430, 4429, 4437
E4S 57820	4427, 4436, 4428, 4439
E4S 58140	4494, 4473, 4495, 4472, 4496



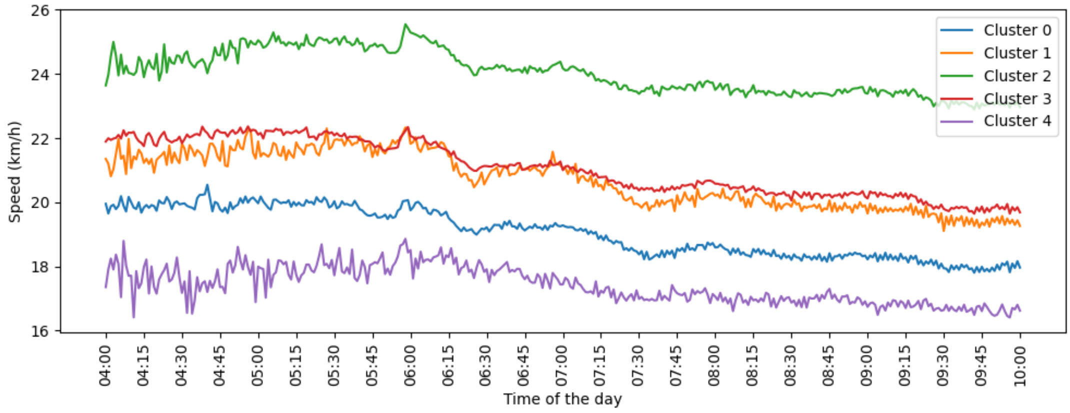
(a) Flow

(b) Speed

Figure B.1: Flow and Speed over the day



(a) Flow



(b) Speed

Figure B.2: Clustering for portals 58140, 57820 and 57435

References

[1] Machine Learning Mastery. Loss function for xgboost.