LUT UNIVERSITY
School of Business and Management
Strategic Finance and Business Analytics

Predicting Short-Term Traffic Speed: A Model Assessment Using Spatiotemporal Variables

2019
Teemu Mankinen
1st Supervisor: Jan Stoklasa
2nd Supervisor: Pasi Luukka

ABSTRACT

Author:	Teemu Mankinen
Title:	Predicting Short-Term Traffic Speed: A Model Assessment Using Spatiotemporal Variables
Faculty:	LUT School of Business and Management
Master's program:	Strategic Finance and Business Analytics
Year:	2019
Master's Thesis:	106 pages, 6 appendices, 26 tables, 25 plots, 10 figures
Examiners:	Post-Doctoral Researcher *Jan Stoklasa* Professor *Pasi Luukka*
Keywords:	Traffic speed, Predicting, Machine learning, Speed drop, Traffic monitoring system, Model comparison, Classification

The focus of the thesis is to examine different machine learning models' ability to predict short-term traffic speed. An autoregressive model, ARIMA model, Linear Regression, K-Nearest Neighbor and Extreme Gradient Boosted Tree are used to predict short-term traffic speed for 5, 10 and 15 minutes forward. Models are compared using root mean squared error (RMSE) and mean absolute percentage error (MAPE) performance measures. Models' performance is also tested with different speed drop levels. Finally, the decision tree algorithm is used to test how well the model is able to classify, if the speed drops below 40 km/h. The results indicated that XGBoost outperforms all the other models in every measurement point and prediction period. The speed drop comparison indicated that the models perform well when the drop is minor but when the magnitude of the drop is large none of the models captures the speed variation desirably. Achieved sensitivity and specificity levels of the decision tree for the 5-, 10- and 15-minute predictions are 0.988, 0.991, 0.995 and 0.524, 0.287, 0.170, respectively. Area Under Curve (AUC) values using the same forecasting periods are 0.962, 0,833 and 0.778.

TIIVISTELMÄ

Tekijä:	Teemu Mankinen
Otsikko:	Liikennenopeuden ennustaminen lyhyellä aikavälillä: Mallien arviointi spatiotemporaalisia muuttujia hyödyntämällä
Akateeminen yksikkö:	LUT School of Business and Management
Maisteriohjelma:	Strategic Finance and Business Analytics
Vuosi:	2019
Pro Gradu:	106 sivua, 6 liitettä, 26 taulukkoa, 25 kuvaajaa, 10 kuviota
Tarkastajat:	Tutkijatohtori *Jan Stoklasa* Professori *Pasi Luukka*
Hakusanat:	Liikennenopeus, Ennustaminen, Koneoppiminen, Nopeuden pudotus, Liikenteen automaattinen mittauspiste, Mallivertailu, Luokittelu

Tutkielmassa tarkastellaan eri koneoppimisen algortimien suoriutumista lyhyen aikavälin nopeuden ennustamiseen. Ennustushorsontti on 5, 10 ja 15 minuuttia käyttäen autoregressiivistä mallia, ARIMA mallia, Lineaarista regressiota, KNN menetelmää ja XGBoost algortimia. Mallien arviointi toteutetaan RMSE ja MAPE suoritusmittareita hyödyntämällä. Valittujen mallien suoritumista testataan myös eri suuruisissa nopeuden pudotuksissa. Lopuksi, päätöspuu algortimilla testataan kuinka hyvin malli kykenee luokittelemaan tilanteita, joissa nopeus putoaa alle 40 km/h. Tuloksien perusteella XGBoost algoritmi pärjää parhaiten jokaisella havaintopisteellä ja ennustehorisontilla. Nopeuden pudotuksien tarkastelussa jokainen malli suoriutuu kiitettävästi tilanteessa, jossa nopeuden pudoitus on pieni. Kun pudotuksen suuruutta kasvatetaan, mikään valituista malleista ei kykene ennustamaan pudotuksia kiitettävästi. Päätöspuun saavuttamamat sensitiivisyys ja spesifisyys tasot 5, 10 ja 15 minuutilla ovat 0.988, 0.991, 0.995 ja 0.524, 0.287, 0.170. AUC arvoiksi samoilla ennustehorisonteilla saadaan 0.962, 0,833 ja 0.778.

ACKNOWLEDGEMENTS

First, I want to thank my supervisor Jan Stoklasa for guiding me through the whole writing process. Your effort was indispensable, and I hope that you keep the same attitude also in the future. A special thanks to Aureolis Oy for providing me the opportunity to finish my studies and supporting the thesis topic.

A huge thanks to my family for supporting me during the whole learning path from the first grade until now. I can conclude now that the hard work pays off! I would also like to give acknowledgements to Roosa for being there when it was needed.

Sincerely,

Teemu Mankinen

TABLE OF CONTENT

APPENDICES

TABLES

PLOTS

FIGURES

LIST OF ABBREVIATIONS

AUC	Area Under the Curve
CV	Cross Validation
FN	False Negative
FP	False Positive
ITS	Intelligent Transportation System
KNN	K-Nearest Neighbor
MAPE	Mean Absolute Percentage Error
RMSE	Root Means Squared Error
ROC	Receiver Operating Characteristics
TMS	Traffic Monitoring System
TN	True Negative
TP	True Positive
XGBoost	Extreme Gradient Boosting

1. INTRODUCTION

The focus of this master's thesis is to examine different machine learning models and their ability to predict short-term traffic speed. An autoregressive model, ARIMA model, linear regression, K-nearest neighbor and Extreme Gradient Boosted Tree are used to predict short-term traffic speed for 5, 10 and 15 minutes forward and the performance of each model are compared using root mean squared error (RMSE) and mean absolute percentage error (MAPE) performance measures. Model performance during speed drops is also examined in the thesis. Selected speed drop levels are < 2 km/h, between 3-5 km/h, 7-10 km/h, 12-15 km/h and when the drop is >= 20 km/h. There are totally three different traffic monitoring system (TMS) points that are used to validate the comparison. In addition, the decision tree classification method is used to test the possibility to predict the speed beyond a certain threshold, which is in this case, average speed less than 40 km/h.

1.1 Background

The precise estimation of travel factors has been in the center of the interest of many researchers for the last decade. While cities are getting bigger also traveling from one place to another gets more difficult. Daily rush hours and unexpected events in traffic will increase travel time, which leads to reduced life quality for people in the cities. (Yildirim & Cataltepe, 2008) Pressure to reduce traffic pollution and save energy acts as a considerable motivator to estimate, not only congestion, but also travel time. Intelligent transportation systems, also known as ITS, include various different technologies that track fluency and safety of the ongoing traffic. Such technologies encase for example mounted traffic sensors, GPS related systems, license-plate recognizers and video cameras (Monahan, 2007). Transparency Market Research (2019) estimates that global ITS market will increase up to 57.44 billion US dollar by the end of 2024 consequently by the desire to protect the environment, save travel time, security and other benefits. As a comparison, in 2015 the corresponding market value was 20.22 billion US dollars. The notable reason

behind the rapid expansion is the increase in demand for traffic control solution and more specific, adaptive signaling systems.

Forecasting short-term attributes such as traffic flow, traffic volume, occupancy, travel time, travel speed, etc. will help develop effective and dynamic control systems for the traffic management and assist them to react in advance to the real-time events on the road. The knowledge of the future occurrences may provide the information to prevent for example congestion which will lead to saving travel time. Ability to predict traffic speed makes possible to calculate how long it would take to move from one place to another which helps to select the shortest way to travel. This will lead to effective time management that is relatively important for all kind of travelers.

1.2 Research objectives and limitations

The focus of the study complies with three different entities that form the disciplines and framework to solve the problem of speed predicting. The literature behind the speed predicting gives the insight of the past and present knowledge around the topic which will work as a guideline how the topic should be and can be approached. After gaining an understanding of the phenomenon and the existing knowledge behind it, it is essential to comprehend the models that can be used for the short-term speed predicting. Five different models are then selected for the comparison of the traffic speed forecasting during different prediction periods. The objective is to examine how well these selected models perform based on the selected performance measures and which model performs the best comparing the others. Another objective of the thesis is also study whether it is possible to predict speed drops under a certain threshold using a classification model. The entities of the speed predicting are presented in Figure 1.

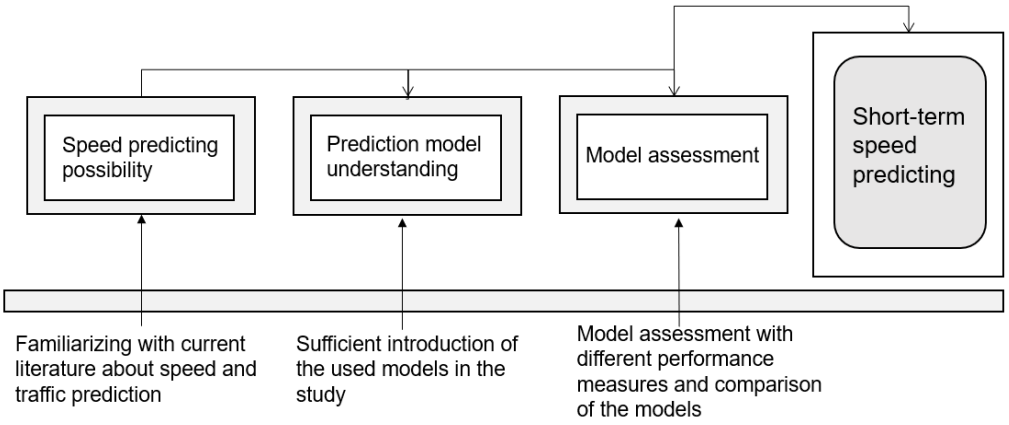


Figure 1. The focus of the study

The research attempts to focus on answering three different research questions that are presented below. In order to achieve the answers, there are certain preparations that are required considering the data. First, before any of the selected models can be evaluated, the raw traffic data must be collected and preprocessed. Predictions are done for three different measurement points that are located near the Helsinki area. Districts where the selected points are located consist of a notable amount of vehicle observations. All three observation measurement points differ from each other based on location, traffic behavior and selected direction of the vehicles. The first two observed points express the variation in traffic speed and flow during the afternoon and morning rush hours. The third observed point provides the aspect of more stable speed during the day.

One considerable limitation of the dataset is the time frame which is obtained from each measurement point. Data is limited to consider only the current traffic behavior of the part of the road that is under investigation, and the selected period is from the beginning of November 2017 until the 27th of September 2018, that will include almost a year of data. Therefore, yearly cycles are not considered in the datasets.

The first research question is:

1. ***Which model accomplishes the best predicting accuracy of the 5-minute average traffic speed?***
   a. *Prediction period 5 minutes forward*
   b. *Prediction period 10 minutes forward*
   c. *Prediction period 15 minutes forward*

In order to answer the first question, the selected models which are AR(1), ARIMA, Linear Regression, K-Nearest Neighbor and Extreme Gradient Boosting are run for all three data measurement points which will lead to running 15 models totally. Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) accuracy measures are used to achieve the answer for the research question.

The second question of the study is:

2. ***How well do the models perform during the abrupt decline of the 5-minute average traffic speed?***

The second question focuses on answering designated models' ability to predict speed drops. The speed drop is tested for the 5-minute predicting period using five different drop levels. The models' forecasting ability is evaluated when the speed drop is < 2 km/h, between 3-5 km/h, 7-10 km/h, 12-15 km/h and when the drop is >= 20 km/h. Drop levels are selected in a way that they cover most of the speed drops in the datasets. The comparison is done by using the same performance measures as stated before.

The third question of the study is:

3. ***Is it possible to predict drops of the 5-minute average speed under 40 kilometers per hour?***

The selected threshold that is used to represent the “jam” condition is 40 km/h. A decision tree is then applied to make the classification, is the average 5-minute traffic speed going to drop under the 40 km/h threshold. The classification is conducted for 5-, 10- and 15-minute prediction periods.

1.3 Structure of the study

The main structure of the study is formalized of 8 different main chapters (Figure 2). The following figure illustrates the structure of the study.

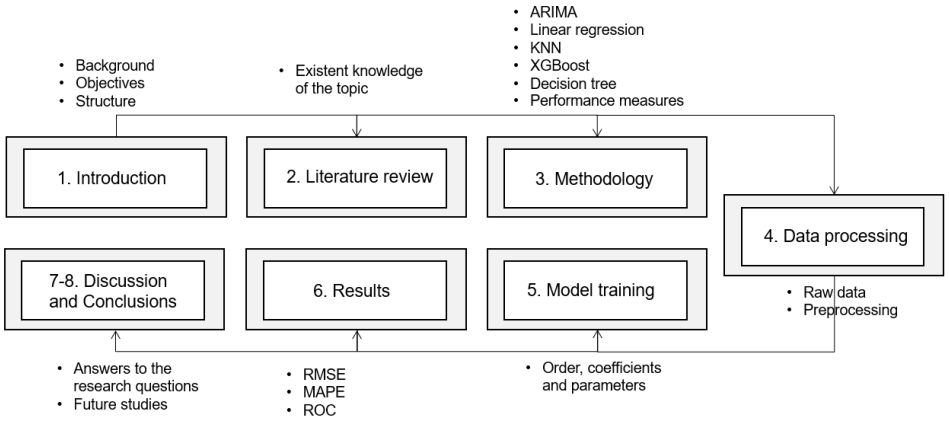


Figure 2. Structure of the study

After the introduction, a brief familiarization with the literature of traffic prediction studies is done. In the section, some of the studies and their results are introduced along with different predicting models that are regularly used to conduct the research of the topic. After the understanding of the literature, theoretical frameworks of different prediction models are introduced. The section focuses on five different models along with the performance measures that are used to compare the accuracy of each used model. This will lead to the part where the data, that is used in the study, is presented and processed to a form that is required to execute the predictions. Furthermore, the model training procedure is done in order to select each prediction model before presenting the results of the study. This includes the determination of the order, coefficients and parameters for the models. The last two chapters summarize the results and conclude the findings of the thesis.

2. LITERATURE REVIEW

Predicting short-term traffic has been a part of Intelligent Transportation Systems (ITS) and related studies since the beginning of the 1980s (Vlahogianni, Karlaftis & Golias, 2014). Accurate predictions are extremely useful for real-time traffic control (Okutani & Stephanes, 1984) which woke up the interest among statisticians and machine learning teams to test and develop various models for forecasting. Most of the studies focus on demonstrating the effectiveness of a certain method under a certain condition using different time frames and roads. That is why it is challenging to find any consensus among the research area because every situation varies from each other significantly.

The prediction period varies from a few seconds to a few hours using current and past traffic data. That creates an interest in the research area to build methodologies which can be used to model traffic volume, speed, travel time and density. The traditional approach is to use classical statistical methods (for example the ARIMA models) for forecasting traffic in a specific point of the road, but lately, more intelligent and data-driven implementations from neural nets to fuzzy-based solutions are established to tackle the problem in question. (Vlahogianni et al., 2014)

A typical characteristic of the traffic prediction is a high observation frequency which gives an excellent framework for testing complex algorithms and their ability to capture the high-intensity variation of traffic. Therefore, traffic predicting can be approached from many different machine learning and statistical viewpoints. Chen, Wang, Li, Hu and Zhang (2012) approached traffic predicting using time-series angle, regression and function approximation-based method were introduced by Dunne & Ghosh (2012), Sun et al. (2012) used the pattern recognition method and Vlahogianni (2009) used a combination of above.

Overall, the literature concludes three different traffic attributes that control almost every study in the area (Figure 3). The framework of the literature balances with selecting the desired parameter to predict which are in most cases either speed, traffic volume or travel time. Between 2004 and 2013 speed was in the center of

interest 24 times, volume 58 times and travel time 40 times (Vlahogianni, Karlaftis & Golias, 2014).

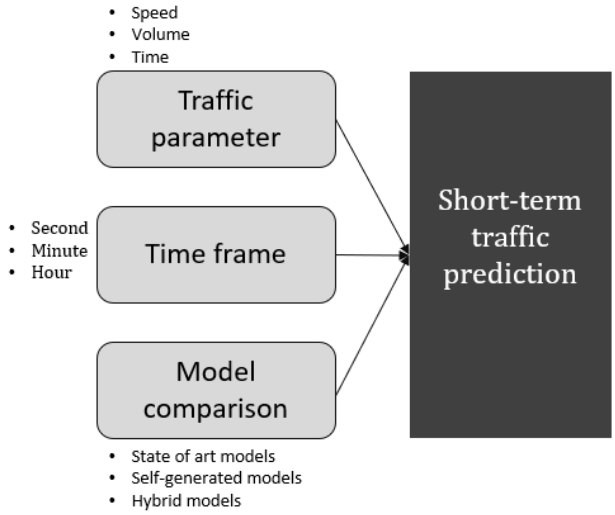


Figure 3. Short-term traffic prediction framework

The second focus in the literature is a time frame which in most cases varies from minutes to hours. The time frame consists of step size and prediction horizon. Prediction horizon defines how many steps forward are tried to predict. Prediction steps, in turn, express the step size which can be for example a minute, 5 minutes or 60 minutes. The last feature is a model comparison, which basically includes comparing the self-optimized or generated models' performance against the state of art approaches. The state of art approaches includes, for example, ARIMA models, support vector regression, linear regression, etc.

2.1 Traffic prediction studies

Yu, Liu, Wu, Liao, Anwar, Li and Zhou (2019) studied short-term traffic speed forecasting and proposed a novel approach to the research area. They studied the influence of different traffic attributes such as traffic speed, occupancy, traffic flow and density of adjacent roads on the traffic speed using real-time data gathered from Hangzhou and Wenzhou. They employed a piecewise correlation function between the traffic attributes and speed and adopted Jenks clustering to determine segment

intervals and compared results to the state of art models. For the 5- and 10-minute time intervals the established Chaos-Wavelet Support Vector Machine model (C-WSVM) produced higher accuracy and stability compared to the state-of-art approaches. Support Vector Machine approach to capture and predict traffic speed was also introduced by Wang and Shi (2010). They stated that one of the biggest challenges considering Support Vector Machines is choosing the right kernel-function for the certain problem that is under investigation. In order to obtain the non-stationary features of the traffic speed data, a new wavelet kernel function was constructed. The results showed that C-WSVM provides feasible results in the short-term traffic speed prediction. Furthermore, the state-of-art Support Vector Machine also performed well in their case.

Cheng, Lu, Peng and Wu (2018) acknowledged the problem of traffic prediction because of the spatial variation of city traffic. They stated that using fixed model structures it is relatively hard to gain solid or satisfactory results. To manage this problem, they proposed an adaptive spatiotemporal K-nearest neighbor model in order to predict short-term traffic. They found out that including the adaptive ability to the K-nearest neighbor model outperforms other models (such as traditional KNN and Elman neural networks) during the compared time periods. Kim, Kim and Ryu (2016) chose the same model approach and studied the goodness of K-nearest neighbor-model to predict traffic speed multiple steps ahead in a different weather condition. The result showed that under a normal weather condition MAPE is less than 6% across every time frame. Under heavy snow MAPE increases to 12 % which indicates that the model captures rapid changes in the weather poorly.

There has been a lot of interest around neural network-based solutions during the past years because of their ability to capture non-linear relationships relatively well. They have multiple applications for example traffic condition forecasting and predicting expected traffic jams (Goves, North, Johnston & Fletcher, 2016). Traffic flow, occupancy and traffic speed forecasting using neural networks is done by Dougherty and Cobbett (1997). Prediction of occupancy and traffic flow using neural networks generated promising results. Although, in case of speed prediction, the model was less successful, presumably because of the effect of slowly moving

vehicles during the low flow conditions. The neural network approach is also implemented by Dunne and Ghosh (2012). They emphasize the demand for applicable multivariate traffic condition forecast model in order to implement adaptive and useful traffic management systems. They propose an artificial neural net (ANN) based model with adaptive learning-based rules predict short-term traffic speed. The result indicated that the implemented forecasting algorithm performs effectively in case of speed and traffic flow prediction. Neural networks were also implemented by Sun, Huang and Gao (2012). They proposed a multi-link model in a case of predicting traffic flows using the past data of the contiguous links. Extracting the relevant information, they combined Graphical Lasso with back propagation neural networks and proved that the constructed model is superior compared to other models in the study.

Traffic-related observations are usually combinations and relations of different attributes between a certain time frame. Predicting traffic occurrences whether it is speed, flow or some other event, includes a time-component that will encourage to use time-series forecasting, either as a benchmark or addition to more sophisticated models. Guo and Williams (2010) argued that predicting short-term traffic state considering the level of traffic and uncertainty, has the possibility to reduce congestion through the traffic operating systems. Regarding the traffic speed, autoregressive moving average and generalized autoregressive conditional heteroskedasticity models were constructed (ARMA and GARCH). ARMA is applied to model speed change, whereas GARCH is used to capture the speed variance. Results indicated that the model performed well in predicting accuracy. Time-series approach for traffic speed predicting was also used by Min & Wynter (2010) for different time intervals. Constructed multivariate spatial-temporal autoregressive moving average model outperformed other published works in 15-minute data and provided decent accuracy with a higher variance 5-minute data.

Tree-based model approaches are also used in the prediction of traffic attributes and they have accomplished a large-scale success in the prediction field overall. Comparing to other machine learning methods that are treated as a black-box, a tree-based approach will provide results that are interpretable and accurate. They

require little preprocessing, can handle various type of predictor variables and can solve complex nonlinear relationships. The tree-based model was used by Zhang & Hagani (2014) to predict travel time in a freeway. They ensemble a gradient boosting regression tree and compared it to other benchmark models such as ARIMA and Random Forest. All three models performed well under a normal traffic condition, but overall, the GBM model performed the best in almost every time steps and segments that were studied. Stochastic Gradient Boosted Decision Tree (GBDT) was also used in order to predict the probability of the incidents (secondary incidents) that occur after the first or the original accident happened. Advanced mathematical methods were applied to predict the incidents that have a low sample mean and small sample size. Bayesian Neural Network was compared to the established GBDT model. According to Park, Haghani, Samuel and Knodler (2018) the methodology that was proposed can have a notable benefit to alert drivers about the expected conditions in a highway.

3. THEORETICAL FRAMEWORKS AND MODELS

The chapter introduces the models and methods that are used in the study. The first introduced model is a linear regression, which is a classic statistical concept to evaluate the relationship between dependent and explanatory variables. From the field of time series forecasting, the basic autoregressive and moving average processes are presented along with the combination of these known as ARMA procedure. From the non-parametric sector, the K-nearest neighbor is introduced and finally, from the boosting algorithms, the Extreme Gradient Boosting is presented. For the final part of the study, the decision tree is used to conduct a classification and therefore it is also presented in this chapter. The chapter familiarizes the concept of cross-validation and also the performance measures used in this study.

3.1 Linear regression

Regression analysis is one of the most widely used methods for analyzing data with multiple factors (Schmidt & Finan, 2018). It is applied for numerous fields in science including economics, physics, engineering and management. Its usability and benefit arise from a logical process to use equation-based interpretation to express the relationship between dependent and independent variables. Popularity behind the linear regression analysis results from relatively simple math and advanced statistical theory. A successful regression analysis includes both theory and practical implementation that are present when the focus is to explain real-life data and situations. (Montgomery, Peck & Vining, 2012, 15, 21)

A simple linear regression assumes the linear relationship between the dependent and independent variable. The general form of simple linear regression is (Schmidt & Finan, 2018) (Moll, Steel & Montgomery, 2016):

$$y_t = \beta_0 + \beta_1 x_t + \epsilon_t \quad (1)$$

where β_0 is the intercept of the equation, β_1 is the coefficient for explanatory variable X and ϵ is the error-term of the model.

A common way to obtain the regression coefficients is to use Ordinary Least Squares (OLS) where the sum of squared differences between the dependent variable and the regression line is minimized. Therefore, the steepness of the regression line is determined by the residuals. Figure 4 presents the regression line between TV advertising and sales using OLS. The grey line represents the residual which is the vertical distance between the regression line and the observation. The response value of the sales is determined by the regression line.

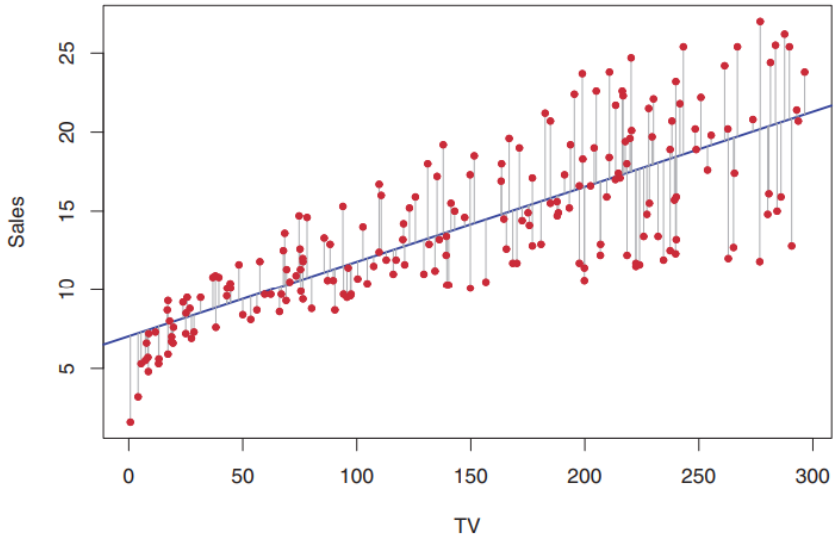


Figure 4. A simple linear regression line between the predictor (TV) and response variable (Sales) (James, Witten, Hastie & Tibshirani, 2013, 61-62).

OLS estimators are gained by minimizing the residual sum of squares $RSS(\beta_0, \beta_1)$ (Weisberg, 2005, 273)

$$RSS(\beta_0, \beta_1) = \sum_{i=1}^{n} [y_i - (\beta_0 + \beta_1 x_i)]^2 \quad (2)$$

where the estimated parameters $\hat{\beta}_1$ and $\hat{\beta}_0$ are (Montgomery et al., 2012, 42):

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} y_i x_i - \frac{(\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i)}{n}}{\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}} \quad (3)$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

A linear regression model that involves more than one explanatory variable is called a multiple linear regression model. Where the simple linear regression fits the regression line between predictor variable and response variable in two-dimensional space, the multiple linear regression fits a surface through a multi-dimensional space of data points depending on the number of explanatory variables. Multiple linear regression is powerful in situations where the linear relationship depends on multiple factors instead of a single predictor. The general form of a multiple linear regression model is as follows (Brooks, 2005, 89) (Lee, Jung & Kim, 2019):

$$y_t = \beta_1 + \beta_2 x_{2t} + \beta_3 x_{3t} + \cdots + \beta_k x_{kt} + \epsilon_t, \quad t = 1,2, \dots, T \quad (4)$$

where β_1 is the intercept of the equation, $x_{2t}, x_{3t}, \dots, x_{kt}$ are explanatory variables, $\beta_2, \dots, \beta_k$ are the coefficients for the explanatory variables and ϵ is the error-term of the model.

3.2 Univariate Time Series Forecasting

Time series forecasting is one of the key machine learning concepts and it has many different implementations. The importance arises from a fact that many prediction problems involve a time component, which is also essential in the case of speed predicting. Time series forecasting can be used to extract statistics and to gain understanding about the characteristics of the data. Using the information, it is possible to conduct precise forecasting results based on the past values and errors of the time series.

3.2.1 Moving Average Process

A simple way to achieve a forecast using time series prediction methodology is to consider past errors as an explaining factor of the future. The Moving Average process is a linear combination of white noise process which relies on previous and current error terms in order to predict the future. It assumes that the current estimation y_t depends on the past values disturbance terms. Constant variance, zero mean and non-zero autocovariances to lag q and zero thereafter are assumed by the model. (Brooks, 2014, 211-212) Let u_t (t = 1, 2, 3…) be a white noise process with

(1) $E(u_t) = 0$

(2) var $(u_t) = \sigma^2$

(3) Covariances y_s

$$= \begin{cases} \left(\theta_s + \theta_{s+1}\theta_1 + \theta_{s+2}\theta_2 \ldots + \theta_q\theta_{q-s}\right)\sigma^2, & s = 1,2,3 \ldots q \\ 0, & s > q \end{cases}$$

The moving average model can be expressed as

$$y_t = \mu + u_t + \emptyset_1 u_{t-1} + \emptyset_2 u_{t-2} + \cdots + \emptyset_q u_{t-q} \quad (5)$$

and using sigma notation

$$y_t = \mu + \sum_{i=1}^{q} \emptyset_i u_{t-i} + u_t \quad (6)$$

where $u_t =$ error term with t-lags. Model is a q^{th} order moving average process denoted as MA(q). (Yabe, 2017) (Brooks, 2014, 211)

3.2.2 Autoregressive Process

Autoregressive models perform under the assumption that past values influence current values plus the error term. Autoregressive models use a combination of previous values of a certain variable to predict future values. It is a widely used method for time series forecasting and for example, parametric spectrum estimation (Kini & Sekhar, 2013). AR(1) indicates a first-order autoregressive model where the first lag of a variable is used to conduct a prediction. Order p autoregressive model *AR(p)* can be expressed as follows (Jung, 2013) (Alvarez-Ramirez & Rodrigues, 2018) (Brooks, 2014, 215)

$$y_t = \mu + \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \cdots + \emptyset_p y_{t-p} + u_t \quad (7)$$

Where u_t is a white noise error-term and y_{t-p} is a previous value of a time series with a lag p. Formula can be expressed more distinctly using Sigma operation.

$$y_t = \mu + \sum_{i=1}^{p} \emptyset_i y_{t-i} + u_t \quad (8)$$

3.2.3 AR(I)MA

Sometimes there is a need to use both, the autoregressive model and the moving average model to conduct a prediction. The combination of previous values and previous errors may generate more genuine predictions compared to the simple autoregressive or moving average models. Combining the autoregressive model and the moving average model together generates an autoregressive moving average model denoted as ARMA(p,q). The model indicates that value y_t depends on its own previous values plus current and previous values of a white noise disturbance term where p stands for the order of autoregression and q the order of moving average. The general equation of the ARMA model can be expressed as follows (Box & Jenkins, 1970, 73) (Brooks, 2014, 224)

$$y_t = \mu + \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \cdots + \emptyset_p y_{t-p} + \emptyset_1 u_{t-1} + \emptyset_2 u_{t-2} + \cdots + \emptyset_q u_{t-q} + u_t \quad (9)$$

For autoregressive moving average models, it is assumed that

$$E(u_t) = 0$$
$$E(u_t^2) = \sigma^2$$
$$E(u_t u_s) = 0, t \neq s$$

For h steps ahead predictions, the process $\{X_t\}$ is defined by (Chevillon & Hendry, 2005)

$$X_{T+h} = \psi^h X_T + \sum_{i=0}^{h-1} \psi^i \varepsilon_{T+h-i} \quad (10)$$

where T is the forecast origin, $X_t = \psi X_{t-1} + \varepsilon_t$, ε is the error and the one-step estimator is determined as

$$\hat{\psi} = argmin_{\psi \subseteq R^{(n+1)\times(n+1)}} \left| \sum_{t=1}^{T} (X_t - \psi X_{t-1})(X_t - \psi X_{t-1})' \right| \quad (11)$$

and the corresponding h -step forecasts are $\hat{X}_{T+h} = \hat{\psi}^h X_T$ with average conditional error

$$E\left[\left(X_{T+h} - \hat{X}_{T+h}\right)|X_T|\right] = (\psi^h - E\left[\hat{\psi}^h\right])X_T \quad (12)$$

Occasionally, there is a need to include integration as a part of the ARMA model. ARIMA is an expansion of the ARMA model and it stands for Autoregressive Integrated Moving Average. Including the integration to a regular ARMA model is usually needed in case of non-stationarity to fulfill the stationarity condition. ARIMA*(p,d,q)* is a general expression of the ARIMA model where p stands for

autoregression, d for the number of integration and q the order of moving average. (Box & Jenkins, 1970, 86-90)

3.2.4 Information Criteria

Selecting an appropriate model to perform prediction is one of the key elements in forecasting. The literature acknowledges various different approaches for dealing with the problem of selecting the optimal model and it is usually dependent on a problem, which is the right one for the specific situation. Two well-known methods are Akaike information criterion (AIC) Bayesian information criterion (BIC), which are measures of the goodness of fit of an evaluated statistical model.

AIC is a method that is used for evaluating the quality of a statistical model. It was proposed by Akaike in 1973 and has expanded by many subsequent studies ever since. The AIC consists of different assumptions considering the data. For instance, It assumes that the data or a group of observations is generated by a random variable which probability distribution is unknown. According to the assumption, the most optimal model is the one where the distance between estimated distribution and real distribution is minimized. (Banks & Joyner, 2017) It tends to provide a sufficient trade-off between the model fitness and complexity that avoids the problem of overfitting. Thus, this makes AIC a good measure comparing other methods to some extent. (Xu, Shao, Qiao & Shang, 2018)

The Bayesian viewpoint is presented in the BIC estimator and it is a part of a criteria class that are dimension consistent. Usually, as in the case of AIC, a model that has the lowest BIC value is preferred compared to others. Overall, the BIC formulation constructs the same way as AIC, but the biggest difference arises from a fact that BIC is providing a uniform estimator of the dimension of the data. (Gkioulekas & Papageorgiou, 2018)

The universal form of AIC and BIC are as follows: (Wagenmakers & Farrell, 2004).

$$AIC_i = -2logL_i + 2V_i \quad (13)$$
$$BIC_i = -2logL_i + V_i log * n$$

Where L_i is the maximum likelihood for the candidate model, i is defined by adjusting V_i free parameters in a way as to maximize the probability that the candidate model has generated the observed data. n is the number of observations that are entered into the likelihood calculation.

3.3 K-Nearest Neighbor

K-nearest neighbor algorithm is one of the widely used and accepted methods in data science, especially in pattern recognition (Gou, Qiu, Yi, Shen, Zhan & Ou, 2019). It can solve various problems regarding classification and regression. It is a non-parametric method that is simple to use and understand. Understandability comes from the fact that there is no need to do any assumptions considering the data. (Kim, et al., 2016) Using a K-nearest neighbor does not require a specific model building for predicting purposes. Instead, it measures a distance between training example and target instance, based on pre-defined nearest neighbor and distance measure to predict the target value. (Kim et al., 2016) Although understandability and usability of the model are simple, it has proved the ability to provide reasonable generalization precision for real-life problems. (Bao, Ishii & Du, 2004)

Defining the optimal K-value has been under discussion for a long time and there is no consensus about the right method to define it. The smaller K-value creates higher noise sensitivity, on the other hand, if the number of K is increased it results as smoother decision boundaries and lower noise sensitivity. Using a high K-value can also lead to class imbalance which can cause problems. Traditionally, the K-value is selected based on test and error. For example, testing the error rate varying different K-values, it is possible to obtain the optimal integer based on the training data. The K-value with the smallest error-rate is then selected for further use. (Ertuğrul & Tağluk, 2017)

The success of the KNN algorithm depends on the number of neighbors selected. The algorithm can be stated as a K-sensitive, which may easily lead to unsuccessful results if the K-value is defined incorrectly. Assuming some positive integer K and a test observation x_o. The algorithm identifies a K-number of integers that are closest to x_o based on a selected distance metric. After that, the KNN algorithm estimates a conditional probability for class j as the fraction of observed points in N_0 whose response value equals j: (James et al., 2015, 39)

$$\Pr(Y = j|X = x_o) = \frac{1}{K}\sum_{i \in N_0} I(y_j = j) \tag{14}$$

Lastly, a Bayes rule is used in order to define the largest probability class for a test observation x_o.

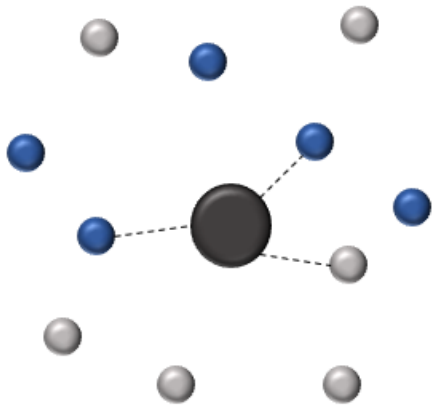


Figure 5. An example of KNN with K=3

Figure 5 illustrates the simple idea behind the algorithm using K=3. Three closest points for the test observation (dark grey circle) are defined based on the distance metric. The predicted value is selected based on the closest points which are, in this case, belong to a blue class.

Although K-nearest neighbor is a method used to solve classification problems, it can be also applied for regression. A common way to implement KNN regression is to calculate the average of the numerical target of the K-nearest neighbors instead of selecting the class based on the neighbors nearby. For given input x of training

data, K observations with x_i that are close are taken into account. The $\bar{y}$ is the average of K independent variables (Goyal, Chandra & Singh, 2014)

$$\bar{y}(x) = \frac{1}{K} \sum_{x_i \in N_{K(x)}} y_i \tag{15}$$

where $N_{K(x)}$ represents K closest points in the neighborhood of x.

There are different ways to measure the distance between train instance and target instance, such as Manhattan distance, the value difference metric and Minkowski distance. The most commonly used distance metrics is Euclidean distance, which can be expressed as the length of the straight line between two points in Euclidean space. The Euclidean distance function between two points it can be expressed as (Wazarkar, Keshavamurthy & Hussain, 2017) (Bao et al.,2004):

$$D(x, y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{16}$$

where x and y are points in m -dimensional space.

3.4 Gradient Boosting

The general phenomenon in a field of machine learning is to achieve and build a non-parametric model from the data to solve regression or classification problem. Although, in real life, possessing a model like this can be extremely difficult or even impossible in some cases. When approaching a problem that is not common, the typical way is to build the model based on a certain theory and adjust the different parameters to achieve better forecasting accuracy. One way to approach data-based modeling is to construct only one strong predictive model. Ensemble methods are also used in modern prediction literature where the final predictive model is

constructed from a set of weak models to gain better forecasting accuracy. (Natekin & Knoll, 2013)

Widely known and used ensemble techniques such as Random Forest depend on a basic averaging of models in the ensemble. Boosting methods, on the other hand, use a different approach to ensemble composition. The basic idea behind the boosting methods is to increment new models to the ensemble consecutively. In every iteration, a new weak base-learner model is trained based on the whole ensemble so far. (Natekin & Knoll, 2013)

One of the successful boosting methods is Gradient boosting or Gradient boosting machines (GBM). In the GBM the learning process sequentially fits new models for achieving better accuracy of the estimate for the response variable. The basic idea behind the algorithm is to build new base learners that are highly correlated with the negative gradient of the loss function. (Natekin & Knoll, 2013) The goal is to find the optimal model that minimizes the loss function (Touzani, Granderson & Fernandes, 2018).

The loss function used in the algorithm is optional and depends on a user, but the classic approach is to use squared errors. The power of the GMB is its flexibility which enables the use of the algorithm in many data-driven situations. In a principle, boosting algorithms are fairly simple to implement which enables the user to try different model approaches. Due to these reasons, solutions generated by GBM has gained a lot of success in different practical problems, data mining and data science competitions. (Natekin & Knoll, 2013)

The classical supervised learning tries to map the $X \rightarrow Y$ relationship based on the training data where X refers to the explanatory variable and Y to the response variable. The data that is used for training the model is $(x_i, y_i)_{i=1}^{N} \subset X \times Y$ where the objective is to establish the functional dependence of $f: X \rightarrow Y$ which is unknown. Y is tried to achieve by minimizing the loss function

$$\hat{f}(x) = argmin_{f(x)} \sum_{i=1}^{N} L\,(y_i, f(x_i)) \tag{17}$$

where y_i is the response variable of x_i. L (.,.) is the loss function which is used to estimate the difference of the true value y_i and the prediction $f(x_i)$. (Xiong, Gui, Hou & Ding, 2018) (Rao, Shi, Rodrigue, Feng, Xia, Elhoseny, Yuan & Gu, 2019)

3.4.1 Extreme Gradient Boosted Tree

Chen and Guestrin (2016) introduced an extended concept of Gradient Boosted Decision Tree (GBDT) for solving real-life regression and classification problems called Extreme Gradient Boosting (XGBoost). One of the significant improvements compared to GBDT is the loss function standardization to alleviate model variances. In addition, it reduces the complexity of the model and the possibility for over-fitness. XGBoost uses the Taylor expansion to enhance the loss function while the classical GBDT addresses only the first derivative in learning. (Chang, Chang & Wu, 2018)

The most meaningful ability the XGBoost has is its scalability for various different situations. The algorithm is built to function even ten times faster than other famous solutions (for example GBDT) using optimization considering the data and calculation. Chen and Guestrin (2016) state that these innovations include:" a novel tree learning algorithm is for handling sparse data; a theoretically justified weighted quantile sketch procedure enables handling instance weights in approximate tree learning. Parallel and distributed computing makes learning faster which enables quicker model exploration."

In the additive learning process of XGBoost, the first learner is fitted to the entire input data. After that, the other model is fitted to the residuals in order to deal with the weak learner failures. The process is looped several times until the stop criterion is fulfilled. The forecasting model is build based on the prediction sums of each learner. In this situation, a weak learner refers to a single decision tree. The common

expression for the prediction at step t is (Fan, Wang, Wu, Zhou, Zhang, Yu, Lu & Xiang, 2018):

$$f_i^{(t)} = \sum_{k=1}^{t} f_k\,(x_i) = f_i^{(t-1)} + f_t(x_i) \tag{18}$$

Where $f_t(x_i)$ is the learner at step t, $f_i^{(t)}$ and $f_i^{(t-1)}$ are forecasts at steps t and $t-1$ and x_i is the input variable.

For handling the problem of overfitting without reducing the calculation speed of the algorithm, the XGBoost model derives the analytic expression below to evaluate the goodness of the model of the original function:

$$Obj^{(t)} = \sum_{k=1}^{n} l(\bar{y}_i\,, y_i) + \sum_{k=1}^{t} \Omega(f_i) \tag{19}$$

Where the n is the number of observations, l is the loss function and Ω is the regularization term expressed as

$$\Omega(f) = \gamma T + \frac{1}{2}\,\lambda||\omega||^2 \tag{20}$$

where λ is the parameter for regularization, ω is the vector of scores in the leaves and γ is the minimum loss that is required to partition the leaf node further. (Fan et al., 2018) (Chen & Guestrin, 2016) (Ma et al., 2018)

3.5 Decision tree

Decision tree refers to a classifier that indicates as a recursive partitioning of the instance space. The basic interpretation of the model is the so-called “rooted tree” that is constructed with a specific number of nodes. In the case of the study, nodes

are either internal nodes that have an outgoing edge or end nodes that are considered as leaves. In the decision tree, the internal node splits the instance space into two or more sub-spaces according to some function related to input variables. The simplest case is when the splitting of the instance space considers only one variable in the internal node. Talking about numeric attributes it is usually referred to a range of values. (Maimon, 2010, 149-150)

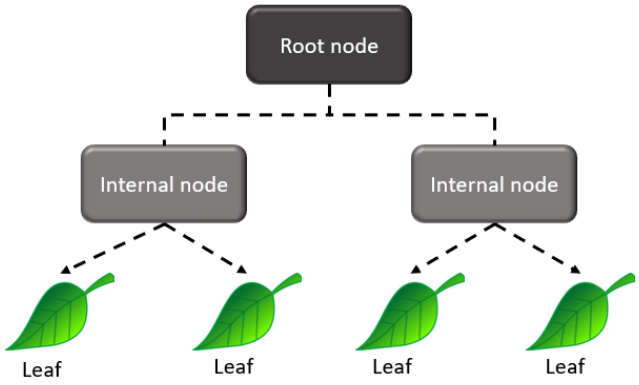


Figure 6. Decision tree example

Figure 6 interprets the basic form of a decision tree. On top of the tree is the root node that refers to the best predictor of the attribute space. The root node has two branches with an internal node and leaf nodes. The internal node further divides to two leaves. In a case of classification, the leaf represents the decision of the algorithm outputs.

For constructing a tree, the hardest part is to define the criteria for splitting the data in each node. Breiman, Friedman, Olshen and Stone (1984) introduced the classical method CART for splitting the data. In order to do the splitting, an impurity function is specified

$$\Delta i(m) = i(m_p) - P_l i(m_i) - P_r i(m_r) \tag{21}$$

where m is the node under testing and m_p is the parent node. P_l is the probability of the object to pass to node m_i (left sub-tree) and P_r is the probability of the object to pass to node m_r (right sub-tree). (Kozak, 2019, 10)

For the decision tree algorithm, the idea is to find the best possible way to split every node solving the maximization problem:

$$argmax_{a_j \leq a_j^R, j=1\ldots,M}[i(m_p) - P_l i(m_l) - P_r i(m_r)] \quad (22)$$

Where M denotes the number of attributes and a_j^R is the best possible division for attribute a_j. A popular method in CART for splitting criteria is to use the Gini index, which is a measure for random variable concentration. The impurity function is then calculated

$$i(m) = \sum_{c \neq o} p(o|m)p(c|m) \quad (23)$$

where $p(c|m)$ is a probability of obtaining decision class c at node m and $p(o|m)$ is a probability of obtaining decision class o at node m. (Breiman et al., 1984, 38) (Kozak, 2019, 10) This leads to solving the formula:

$$argmax_{a_j \leq a_j^R, j=1\ldots,M}(-\sum_{c=1}^{C} p^2(c|m_p) + P_l \sum_{c=1}^{C} p^2(c|m_l) + P_r \sum_{c=1}^{C} p^2(c|m_r)) \quad (24)$$

where $p(c|m_p)$ is the probability of obtaining decision class c at node m_p, $p(c|m_l)$ is the probability of obtaining decision class c at node m_l, $p(c|m_r)$ is the probability of obtaining decision class c at node m_r and C is the number of classes (Kozak, 2019, 10).

3.6 Cross-validation

Choosing the right model is an indispensable part of the process of developing practical prediction models. Numerous amounts of papers have discussed different ways to select the right and useful model for different practical and mathematical situations and they tend to answer the question: How to select a functional modeling

operation that normally includes the selection of the model and parameter estimation? In normal circumstances, it can be extremely difficult to know which procedure works best for the selected data. (Zhang & Yang, 2015)

At the beginning of the 30s, it was already pointed out that using the same data to train and evaluate algorithm generates overoptimistic results. It raised the need to develop methods that can overcome the problem of overfitting. In the classic data science, data is split to train set that is used to train the algorithm and to test set that will work as a validation set for the trained model. The validation sample takes the role of a “previously unseen data” that will tell how well the trained model performs. (Arlot & Celisse, 2010)

In a case of regression or classification, one of the most widely used method to assess the generality of an algorithm is cross-validation (Bergmeir, Hyndman & Koo, 2018). In cross-validation, each part of the training data is used to train the model and the test set is utilized for validating the trained model. The reason behind cross-validation is that it avoids overfitting that can be a problem while choosing the right parameters. Due to this, cross-validation has used extensively in a field of data mining, especially in the model selection. (Zhang & Yang, 2015)

3.6.1 K-fold cross-validation

Even though there are different cross-validation techniques, the most classical way to conduct cross-validation is to perform Leave-N-out CV procedure, but there are also methods such as Monte-Carlo CV, Generalized CV and Repeated CV available. In Addison to these, K-fold cross-validation is widely used in the research field. The general idea behind K-fold CV is to divide a training set into K-number of folds, which each contains a training set and a validation set. (Jung & Hu, 2015) The selected model is trained using the training set and is then fitted to the remaining K-1 folds. Mean squared error (MSE) is calculated on the remaining held-out fold. The procedure is repeated K-times and the K-fold cross-validation estimate is obtained by averaging MSE values. CV estimate can be illustrated as

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_k \tag{25}$$

where K is the number of folds and MSE is the mean squared error of each fold. (James et al., 2013, 176,179)

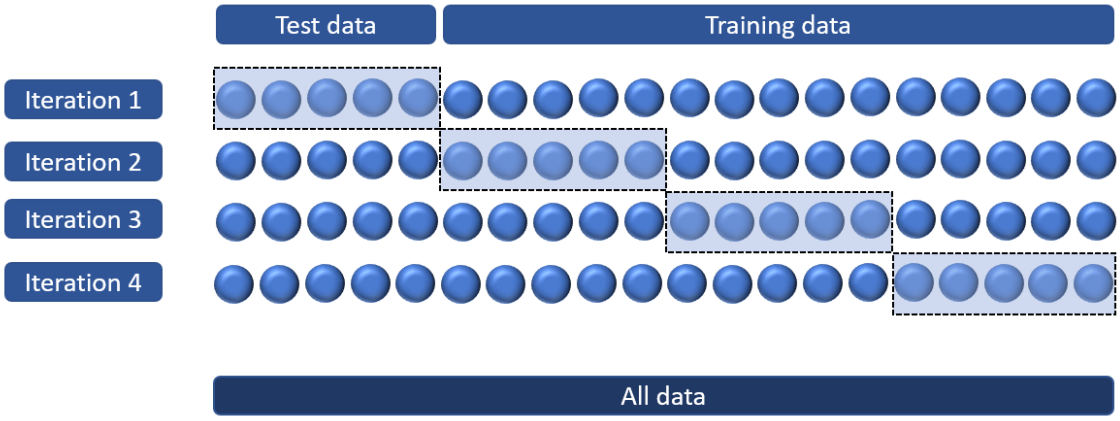


Figure 7. 4-fold cross validation example

Figure 7 visualizes the idea behind 4-fold cross-validation. Each of the 5-point cluster is used as a test set for the rest of the observations that generate the training set. The procedure is repeated K-times which is in this case, four times.

3.7 Model Performance

In data science, there is an incentive to find the right measure to capture the performance of the models. The field of statistics acknowledges many different ways to evaluate the goodness of the model for example based on the interpretation of the average error or percentage of the errors. In a case of selecting time series model, AIC and BIC that were discussed before, work as a relative model fit indexes and are often used for model comparing. Nevertheless, they do not provide knowledge about absolute model fit that can be in some cases, more practical and informative. (Schubert, Hagemann, Voss & Bergmann, 2017)

3.7.1 Root Mean Squared Error

One of the most frequently implemented model performance measures is Root Mean Squared Error, which can be stated as a root squared difference between predicted and actual values divided with the number of observations. It is a measure of mean deviation that resembles standard deviation and works relatively well as a general error metric. In the literature, it is applied to measure the performance of the model in the area of meteorology, traffic forecasting, finance and many others. (Chai, Draxler, 2014) (Vlahogianni, 2015) (Aslanidis, Christiansen & Cipollini, 2018) RMSE should be considered as a relative measure while comparing different predictions and used for the equal data series among different models. A small measure value indicates good forecasting ability and model with the smallest RMSE value is normally the one that is selected. The general form of the RMSE can be expressed in the case of the study as (Yu et al., 2019)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(as_i - ps_i)^2}{N}} \tag{26}$$

where as_i is the actual traffic speed, ps_i is the predicted traffic speed and N is the number of observations.

3.7.2 Mean Absolute Percentage Error

Another popular method for measuring the goodness of fit of a model is Mean Absolute Percentage Error (MAPE) and it is popular in practical cases (McKenzie, 2011). There is a vast range of applications that fit well for MAPE and it is argued to be good in a case of real-world applications where actual values stay above zero. The disadvantage of the measure is that if the actual value is zero, the measure can be infinite or undefined and when close to zero, the distribution can be highly skewed (Hyndman & Koehler, 2006). Due to non-negativity, it is broadly applied in a field of economics, for example, predicting sales in a monthly or quarterly basis (Ren & Glasure, 2009). It is also very popular for predicting applications where a sufficient

amount of data is available (Myttenaere, Golden, Le Grand & Rossi, 2016). Like using RMSE, a model that has the smallest MAPE value is usually selected. MAPE informs the accuracy of the models as a percentage. The mathematical form of MAPE is (Yu et al., 2019)

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|as_i - ps_i|}{as_i} \times 100\% \quad (27)$$

where as_i is the actual traffic speed ($as_i \neq 0$), ps_i is the predicted traffic speed and N is the number of observations.

3.7.3 Receiver Operating Characteristic, Accuracy, Sensitivity and Specificity

A confusion matrix simplifies the interpretation of the actual and predicted values in a case of classification. For a k-class classification, the confusion matrix is $k * k$ contingency table with cells$[i, j](i = 1, \dots, k, j = 1, \dots, k)$ (Ruuska, Hämäläinen, Kajava, Mughal, Matilainen & Mononen, 2018). A general form of a binary class confusion matrix is as follows:

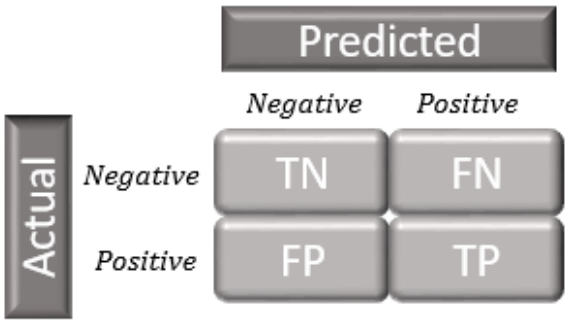


Figure 8. Confusion matrix for a binary classifier

In the confusion matrix, section TN (True negative) is the amount of correct prediction where the instance is negative, FN (False negative) refers to incorrect predictions in a case where instance is positive, FP (False positive) in turn, is a section that tells the number of incorrect predictions in a negative instance case.

Section TP (True positive) holds the correct values when the instance is positive. (Santra & Christy, 2012)

Model accuracy, sensitivity and specificity are popular methods to gain the understanding of the classification model's ability to classify test observations. Concerned measures are obtained based on the confusion matrix instances. The accuracy measures the sufficiency of the model overall. Sensitivity, also referred to as true positive rate, measures the proportion of true positives that are correctly predicted. Specificity, in turn, is referred to as true negative rate that tells how well the model classified true negative values. Accuracy, sensitivity and specificity are calculated the following way (Brzezinski, Stefanowski, Susmaga & Szczech, 2018).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{28}$$

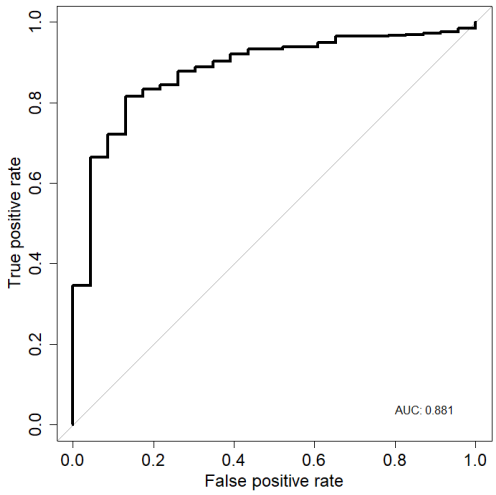
$$Sensitivity = \frac{TP}{TP + FN} \tag{29}$$

$$Specificity = \frac{TN}{TN + FP} \tag{30}$$

Derived from the sensitivity and specificity, the Receiver Operating Characteristic (ROC) curve is one of the most common statistical tool to evaluate classifier performance (Gigliarano, Figini & Muliere, 2014). ROC curve expresses the trade-off between true positives and false negatives for all viable thresholds (Cook, 2017). It is described as a plot of $Se(c)$ versus $1 - Sp(c)$ for $-\infty \leq c \leq \infty$, or comparably as a plot of

$$ROC(t) = 1 - G\left(F^{-1}(1 - t)\right) \tag{31}$$

over $t \in [0,1]$, where $F^{-1}(1 - \mathrm{t}) = inf\{x \in R: F(x) \geq 1 - t\}$ (Goncalves, Subtil, Oliveira & Bermudez, 2014).



Plot 1. ROC curve example

The diagonal line represents the random selection for a classifier. When the curve trends in the left upper side of the plot, it usually indicates that the model performs better than random selection. More upper left corner the curve gets, better the trade-off between sensitivity and specificity is. The false positive rate is expressed as $1 - specificity$.

The area under the curve (AUC) is the measure of the area under the ROC curve which can be used to rank the performance among different classifiers without requiring a specific threshold. AUC is a widely used measure especially in a situation where the classes are imbalanced (Cheng, Fu & Qiu, 2019) and a high AUC value usually denotes that the model performs the classification task well. When the decision threshold is varied one way to obtain the AUC value is to use trapezoidal integration which is

$$AUC = \sum_i \left((1 - \beta_i \times \Delta\alpha) + \frac{1}{2}(\Delta(1-\beta) \times \Delta\alpha) \right) \tag{32}$$

where

$$\Delta(1-\beta) = (1-\beta_i) - (1-\beta_{i-1}),$$
$$\Delta\alpha = \alpha_i - \alpha_{i-1}$$

and $1 - \beta$ is a true positive rate and α is a false positive rate and using threshold level i. (Bradley, 1997)

4. DATA

The data that is used in the study is gathered from two different open source platforms: The Finnish Transport Agency and The Finnish Meteorological Institute. There are totally 26 different explanatory variables in the dataset which are used to explain the traffic speed for 5, 10 and 15 minutes forward. Variables are either engineered derivatives from other variables or aggregated 5-minute averages from the raw data. First, the characteristics of the raw traffic data are introduced where the vehicle class variable is treated as a binary variable. After the raw traffic data, the weather variables are presented and visualized.

4.1 Raw traffic data

In the thesis, the traffic data is gathered from the website of the Finnish Transport Agency that has published the traffic-related information from the year 1995 forward. The Finnish Transport Agency collects the data about road traffic using an automatic traffic monitoring system (TMS). TMS points functionality is based on induction and change in a loop's inductance. The loop technique is installed inside the road pavement and when the vehicle crosses the loop, the record is stored. There are currently around 500 different traffic measurement stations in Finland. (The Finnish Transport Agency, 2018a) TMS points gather the following information of the bypassing vehicle:

Table 1. Raw data variables

Variable	Value
TMS point	e.g. 149
Year	*18*
Running day number	*1-366*
Hour	*0-23*
Minute	*0-59*
Second	*0-59*
1/100 Second	*0-99*
Vehicle length	*m*
Lane	*1-?*
Direction	*1 or 2*
Vehicle class	*1-7*
Speed	*km/h*
Faulty	*0-1*
Total time	*-*
Time interval	*-*
Queue start	*-*

Lane-variable expresses the specific lane that a crossing vehicle uses. It can vary according to the direction of the vehicle or the number of lines in a certain part of the road. Direction indicates which way the vehicle was driving. For example, if a car crosses point 149 at Kehä 1 and the recorded direction is 2, it indicates that the car was driving towards Tapiola. There are seven different vehicle categories depending on characteristics of the surpassing vehicle. Different categories are expressed in Table 2

Table 2. Vehicle classes

Category	Vehicle
1	*Car or van*
2	*Truck*
3	*Bus*
4	*Truck with semi-trailer*
5	*Truck with trailer*
6	*Car or van with a trailer*
7	*Car or van with a caravan or long trailer*

Faulty variable indicates whether there was an error considering the observation TMS point recorded. There are several rules that define the faultiness of the observation (Finnish Transportation Agency, 2018b). Rules are listed in Table 3:

Table 3. Faulty conditions

Faulty rules
Year < 0 or Year > 99
Day < 1 or Day > 366
Hour < 0 or Hour > 23
Minute < 0 or Minute > 59
Second < 0 or Second > 59
1/100 Second < 0 or 1/100 Second > 99
Speed < 2 or Speed >= 199
Direction < 1 or Direction > 2
Vehicle class < 1 or Vehicle class > 7
Lane < 1
Length <= 1 or Length > 39.8

If one of these conditions is true observation gets Faulty value 1, otherwise, the value will be 0. Total time, Time interval and Queue start are technical indicators of the observation and are not used as an explanatory variable. In addition, the Lane variable and 1/100 second are not used in further analysis.

4.2 Data points

There are totally six different TMS point's data that are required and therefore obtained. The selected points are 149, 148, 107, 4, 126 and 145 where 149, 107 and 126 will act as main data points for average speed predicting. Information gathered from 148, 4, and 145 will serve as a previous data point for the main TMS points. The three main observation points are selected in order to see the possible changes in model prediction accuracies when the traffic dynamic varies. Previous TMS points will give information about the traffic state before the main point. Figure

9 includes all the points used in the thesis. The example of the raw data from point 107 is expressed in Appendix 1. The column indicators are the same as in Table 1. The data is gathered from the beginning of November in 2017 to 27th of September in 2018 and the time period is the same for all TMS points.

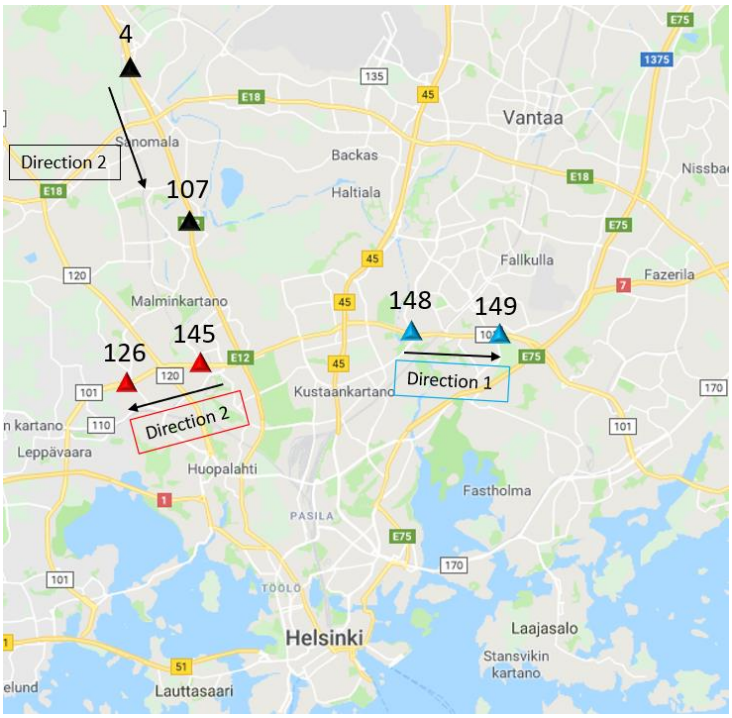


Figure 9. Data point locations and directions

4.3 Data preprocessing

In order to have the data in a form that is required to conduct the analysis and run the models, the data needs to be preprocessed. It is a vital part of predicting process and the focus is to gain the final dataset or datasets that are usable for further analysis (García, Luengo & Herrera, 2016). Furthermore, an increasing amount of data requires essential techniques for data reduction to ease the data mining process (Alvarez-Ramirez & Rodriges, 2018).

First, the data has to be filtered based on the direction of the vehicles. For the TMS point 149, the selected direction of the vehicles is from West to East which signifies the direction from Tapiola to Itäkeskus. Data is filtered that only observations where

direction 1 is fulfilled are selected. Chosen direction for the other two points is 2 which can be discovered from Figure 9. TMS points 149 and 107 directions express the variation in traffic speed and flow during the afternoon and morning rush hours. The third TMS point (126) direction provides the aspect of more stable speed during the day. The data amounts after filtering are as follows:

Table 4. Amount of observations

TMS point	*Direction 1*	*Direction 2*	*Total*	*Direction 1 or 2 after faulty values*
149	*10377205*	*11012437*	*21389642*	*10032416*
148	*10151242*	*10367144*	*20518386*	*10142228*
107	*8387710*	*8522566*	*16910276*	*8479174*
4	*9220268*	*9366276*	*18586544*	*9356928*
126	*12562153*	*12938623*	*25500776*	*12809634*
145	*14387642*	*14891304*	*29278946*	*14836635*

The green numbers are values of the direction that is examined at each point. There are slightly over 10 million observations in 149 and 148 whereas in 107 and 4 the number of vehicles during the time period is 8.5 million and 9.36 million. The greatest amount of records, 14.9 million, was gathered from point 145.

After reducing the data using only the desired directions, faulty values are omitted from the dataset. Removing faulty values is necessary because there is no guarantee about the validity of these observations. Leaving these records to the dataset can have an undesirable effect on the used models and results. That is sufficient reason to extract all the records from the dataset that are faulty. The amount of data observations after filtering the faulty records are also expressed in Table 4.

4.3.1 Data aggregation and missing value imputation

The speed of the vehicles is aggregated to the 5-minute average which means averaging the speed of all cars that have crossed the TMS point during the five-minute period. For the count variable and vehicle variables, the value is the sum of each category vehicle during the 5-minute time frame. To calculate the average of each 5-minute period, the simple formula of average is used:

$$Average = \frac{1}{n}\sum_{i=1}^{n} obs_i \tag{33}$$

where n is the number of observations which is in this case a number of crossing vehicles during the 5-minute period and obs_i is a single vehicle. After the aggregation, the data amount reduces to 95313, 95312 and 95311 observations during the 5-, -10 and 15-minute time periods.

Generally, predictive models receive data from different sensors in the case of traffic speed prediction. A high-volume data inevitably suffers from noise or in some situations, the data is not available at all. Turner, Lomax & Margiotta (2000) showed in their study that the percentage of missing data varied from 7 to 94 in the sensors of 23 cities under investigation. This states that missing data is clearly a notable problem in time management systems. (Smith, Scherer & Conklin, 2003) Missing values can occur for several different reasons, for example, because of the malfunction of a sensor or due to transmitting error. (Chen, Grant-Muller, Mussone & Montgomery, 2001) Depending on a situation, filling the missing values can be necessary for the prediction models and there are several ways to do it from simple models to more complex ones that consider various spatial-temporal relationships, e.g. the average values from other sensors nearby. (Laña, Olabarrieta, Velez & Del Ser, 2018)

Various reasons can cause the missing value in the data, but in the case of the study, there are three excuses behind them that are most likely. One of the obvious

reasons is that there is no car crossing the point during the 5-minute period. Situations like this occur most likely during the night time because the amount of traffic is substantially lower compared to the day time. As an example, the average amount of cars in every five minutes from 12 a.m. to 6 a.m. at point 149 during the observation period is 28 and between 12 p.m. and 6 p.m. 180. The second reason for the missing values can be roadworks. Roadworks can cause the part of the road to be unusable for several days in a row, which means that no observations are recorded during that time. The third reason can be a faulty recording device or transmitting problem as mentioned before.

Point 149 has totally 319 missing values during the observation period. Most of these observations lack because there were no recordings between 18.9.2018 and 19.8.2018 afternoon. This makes totally 173 missing values. For points 148, 107, 4, 126 and 145 the number of missing values in the dataset are as follows: 232, 95, 47, 26 and 702, respectively. The amount of missing values from point 145 comes from the fact that there were no observations between 18.12.2017 and 20.12.2017 9:50.

There are various missing value imputation methods available to deal with the problem. It is important to emphasize that there is no supreme way to deal with the problem that outperforms the others. After all, it is about making an approximation of the data that is as close as possible for reality. In this case, missing values are imputed using the past 15-day averages on that exact minute in order to get a complete time series. For example, if there is a missing value at 16.5.2018 00:00, the filled value is the average of the past 15 days at time 00:00. Leaving the missing values unfilled could lead to unreliable results because there are large gaps in the datasets that are used.

4.4 Weather data

Weather data is added as an explanatory variable to capture the uncertainty generated by changing weather conditions. A considerable amount of studies

focuses on the effect of weather conditions and more precisely, extreme weather condition to the traffic flow. For example, high-intensity rain can have stern effects to the transportation (Guo, Wu, Tong, Zeng, Yang, Chen, Zhu & Li, 2018) Sathiaraj, Punkasem, Wang and Seedah (2018) proved that for example visibility and temperature attributes have a significant effect on traffic volume. Additionally, considering lethal accidents, weather data helps to improve crash risk analysis in a cost-effective way. (Chung, Abdel-Aty & Lee, 2018)

The data is gathered from the Finnish Meteorological Institute using their open source platform. The open-source platform provides support for different research projects, cooperation and business for international and local use. (Finnish Meteorological Institute, 2018) Used variables and their metrics are as follows:

Table 5. Weather variable metrics

Variable	Metric
Rain intensity	*mm*
Snow debt	*mm*
Temperature	*Celsius*
Visibility	*m*
Wind speed	*m/s*

The data is gathered from Kumpula weather station located in Helsinki. The information about the weather conditions is provided in every 10 minutes. Because the traffic data is aggregated to the 5-minute averages, the missing 5-minute gap is imputed with the previous 10-minute predictions considering the weather conditions. Although the last 5-minute values might not reflect the true values of the future, they reflect the reality in a reasonable accuracy. Plot 2 visualizes all weather variables that are used in all three datasets. The statistics of each variable are expressed in tables 7, 8 and 9 with all other continuous explanatory variables.

Plot 2. Weather data during the estimation period

4.5 Variables

In a comparison of different models in the thesis, totally 26 different explanatory variables are used. The focus is to explain the speed 5, 10 and 15 minutes forward

in order to see how well the speed and the speed drops can be predicted with selected models. Variables that are used in the analysis after aggregation and engineering are as follows.

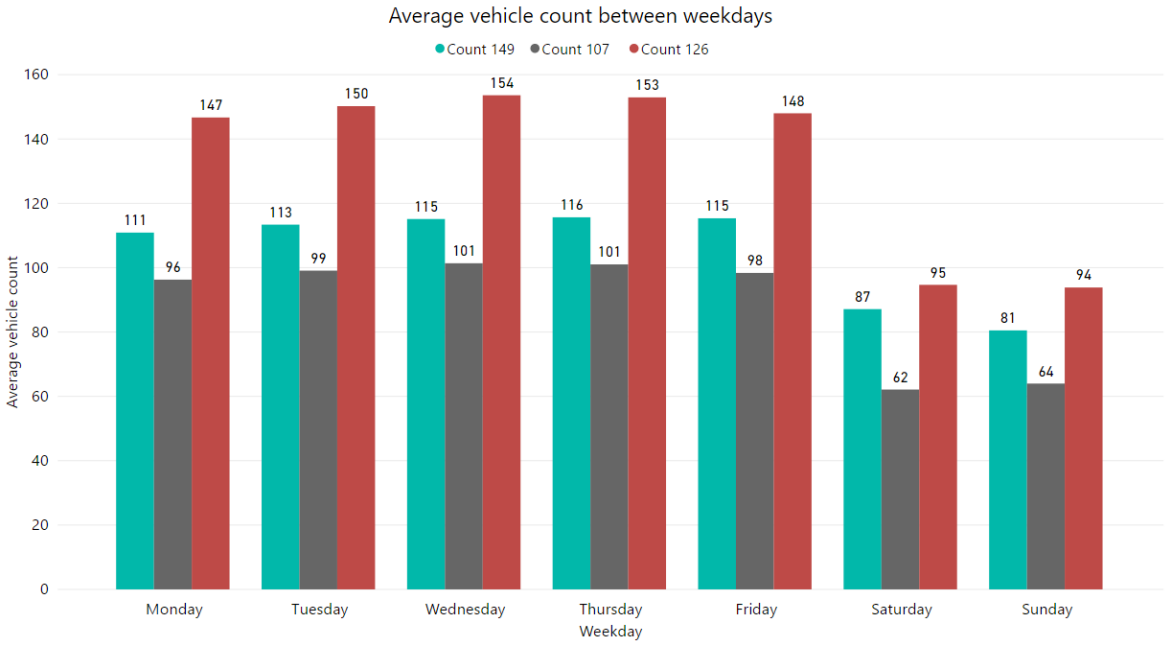
Table 6. Explanatory variables

Variable	Category
Speed lag 1	*Continuous*
Speed lag 2	*Continuous*
Speed lag 3	*Continuous*
Speed lag previous TMS point	*Continuous*
Count lag 1	*Continuous*
Count lag 2	*Continuous*
Count lag 3	*Continuous*
Count lag previous TMS point	*Continuous*
Vehicle class 1-7	*Integer*
Hour	*Integer*
Minute	*Integer*
Morning time	*Binary*
Afternoon time	*Binary*
Weekend	*Binary*
Holiday	*Binary*
Rain	*Continuous*
Snow	*Continuous*
Temperature	*Continuous*
Visibility	*Continuous*
Wind speed	*Continuous*

The autoregression of the predicted speed is considered using three different speed lags. The speed lag 1 variable is the average speed 5 minutes ago whereas the speed lag 2 and 3 indicate lagged values for 10 and 15 minutes ago. The previous TMS point speed lag expresses the TMS point record on the same road before the main point. For point 149 the previous TMS is 148, for the 107 the previous point is 4 and for the 126 the previous point is 145 as Figure 9 indicates. Lagged value of the previous point is the last 5-minute average. Count lags work as the same principle as the speed lags. Count lag 1 is the last 5-minute average, count lag 2 is the average 10 minutes ago and count lag 3 is the average 15 minutes ago. Previous TMS point count lag is the last 5-minute count from the previous TMS point. Vehicle class consists of seven different variables. It addresses how many times a certain

category vehicle has crossed the TMS point. There are totally seven different vehicle classes, which creates a totally seven different explanatory variables (Table 2).

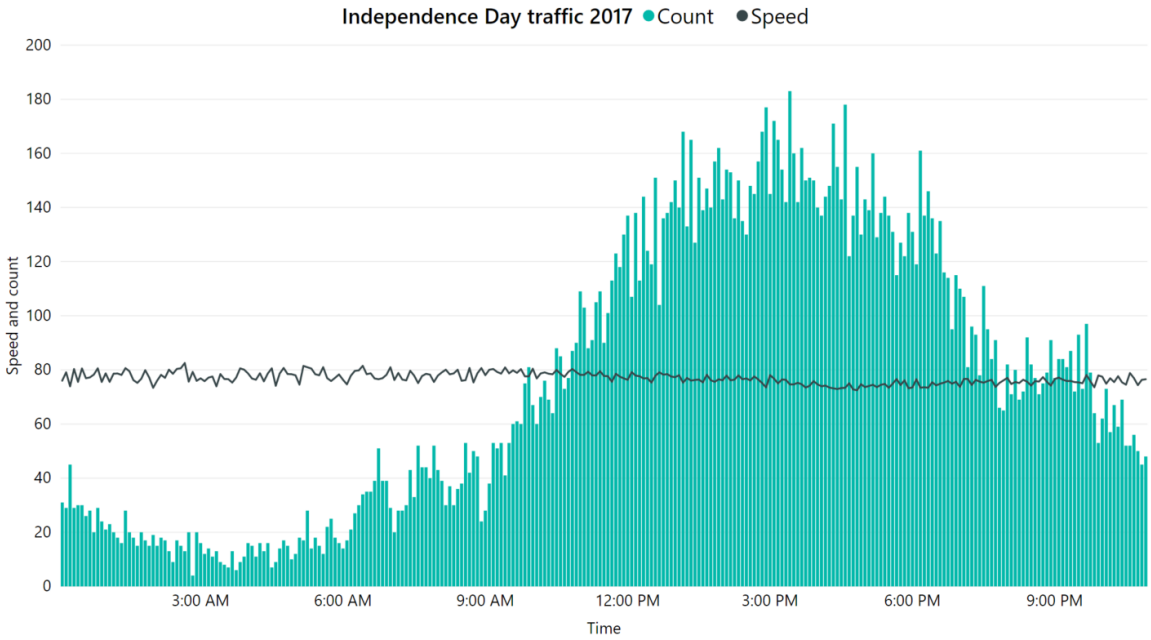
The hour variable is an integer and indicates the hour of the day from 0 to 23. A minute, in turn, indicates the minute of the hour varying from 0 to 55 in 5-minute steps. Morning time and afternoon time variables are constructed as a binary variable that indicates the morning and afternoon increase in traffic flow and a possible decrease in traffic speed. The morning time gets value 1 when the hour is between 7 and 9 and the afternoon time when the hour is between 15 and 18. In Plot 3, one can see the average traffic amount during the weekdays. As Plot 3 indicates, there is a clear decrease in traffic amount during the weekends.



Plot 3. The average traffic speed and count by weekday

The weekday dimension is included using a binary variable that gets value 0 when the day is a normal weekday and 1 when it is weekend. This is mainly due to the reason that traffic works differently during the week because people are traveling to work. During the weekends the traffic is lower and more stable. The holiday variable is also included as an explanatory factor. Holiday variable gets value 1 if it is

included to the national holidays in the Finnish calendar. The holidays during the observation period are listed in Appendix 2.



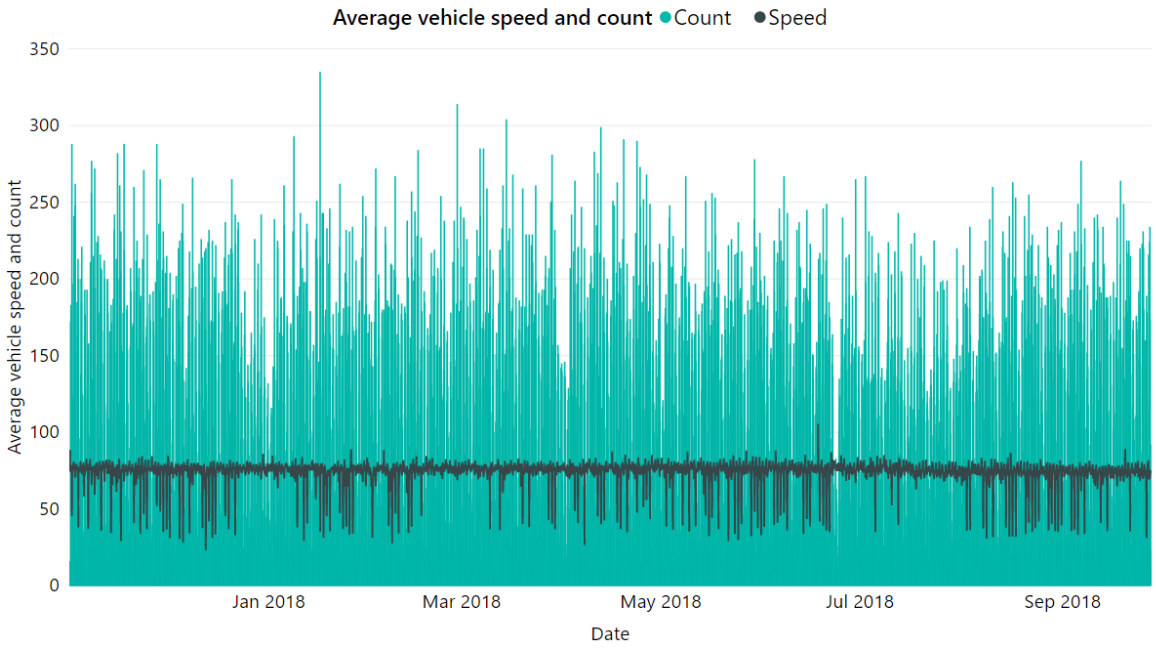
Plot 4. Speed and count during the Independence Day 2017 at the TMS point 149

As an example, the speed and count from TMS 149 during the Finnish Independence Day 2017 are plotted in Plot 4. Speed stays constant during the day even though the amount of traffic increases substantially.

4.6 TMS point 149

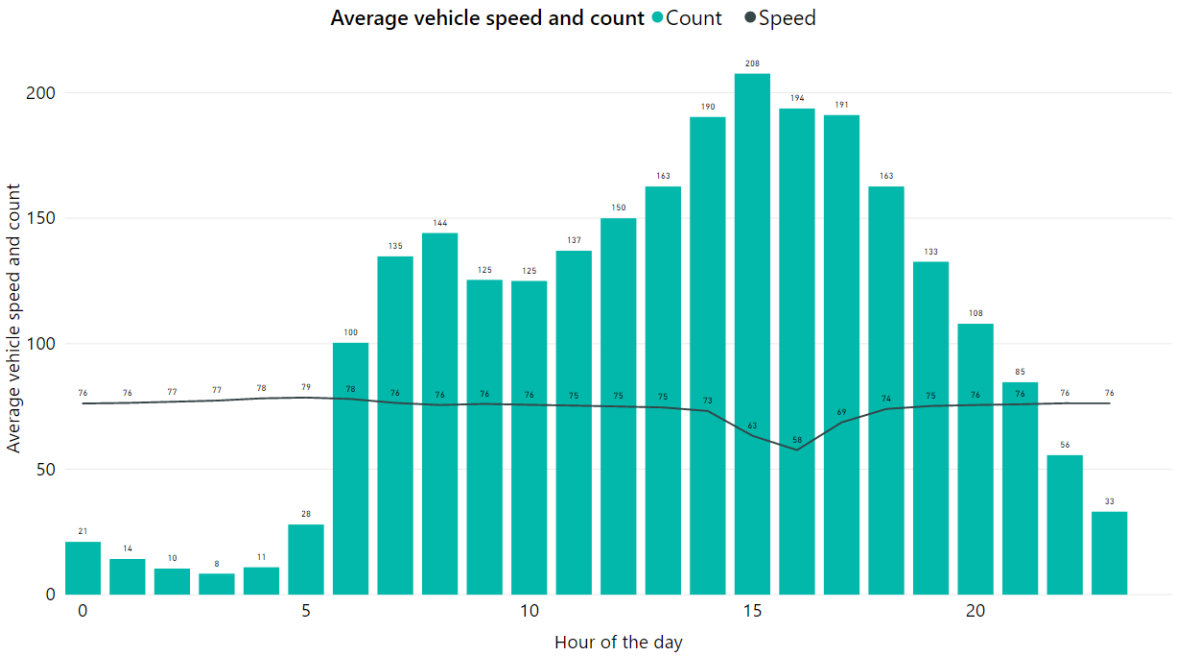
Plot 5 expresses the time series of the speed and count of the vehicles at point 149. As the plot denotes, there is a variation between the average speed and count to some extent. Grey spikes in the plot indicate that either a sharp drop or increase in the average vehicle speed has occurred. Smallest speed during the observation period took place 12.12.2017 at 16.00 and the greatest 18.6.2018 at 03.15. The smallest recorded average speed is 23.6 km/h and the largest 105 km/h. The mean speed is 74.4 km/h and the standard deviation of 7.6 km/h. The largest count during the observation period is 335 vehicles. After the missing value imputation, the smallest number of crossing vehicles is 1, even though it is possible that there are

times when no car has crossed the measuring point. Mean count in the dataset is 105.5 and the standard deviation is 74.4. Other statistical measures of the speed and count at point 149 can be found from Table 7.



Plot 5. TMS 149 speed and count

The average speed and count by the hour of the day are plotted in Plot 6. The average speed stays almost constant during the day. Between 15 and 17 the average speed tends to drop on average, and the smallest average speed occurs during the 16 (58 km/h). This is due to the reason that people are leaving work during that time. Speed drop on average seems to correlate with the high number of vehicles. As the plot indicates, the highest amounts of vehicles occur between 14 and 17, which is around the same where the average speed drop occurs.



Plot 6. The average vehicle speed and count by the hour of the day at point 149

Table 7 presents statistical measures of different continuous explanatory variables and the independent variable speed. The previous TMS point has the highest and the lowest speed values of 110 km/h and 10 km/h, whereas the highest count is 352. Vehicle class 1 (car or van) is the largest of the vehicle classes with over 9 million observations during the time period. The second largest vehicle group is vehicle 2 (trucks) with slightly over 200 thousand observations. The vehicle class 7 (Car or van with a caravan or long trailer) is the smallest one with around 44 thousand observations. Rest of the vehicle classes and their sums are found in Table 7.

Weather statistics that were plotted before are also included in the table. The maximum amount of rain is 6.5 mm, whereas the maximum amount of snow reached 30 mm. The temperature increased to 32 degrees at the highest and the lowest -20 degrees. Visibility dropped to lowest (140 meters) in 4th of April 2018 and the maximum recorded wind speed in the observation point was 14.7 m/s.

Table 7. TMS 149 variable statistics

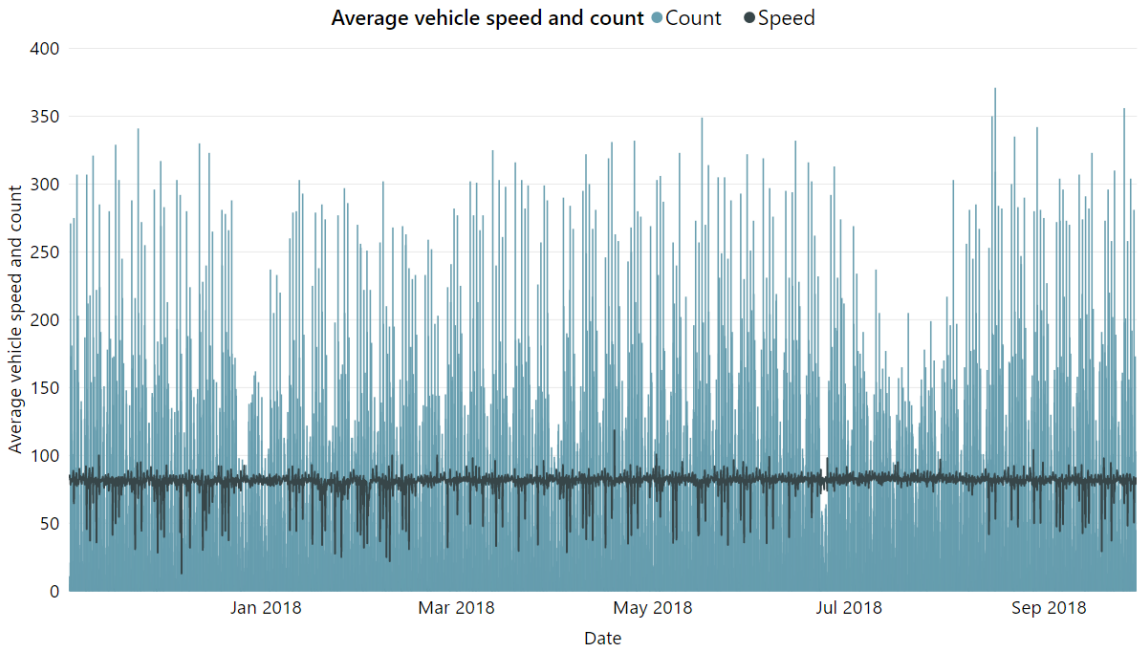
Variable	Mean	Median	Mode	Standard Deviation	Kurtosis	Skewness	Min	Max	Sum
Speed	74.4	75.8	77.0	7.6	16.6	-3.9	23.6	105.0	-
Speed previous TMS point	78.2	78.9	80.0	6.0	56.6	-6.8	10.0	110.0	-
Count	105.5	115.0	9.0	74.4	-1.1	0.2	1.0	335.0	10055158
Count previous TMS point	106.6	113.0	8.0	78.0	-1.0	0.3	1.0	352.0	10158543
Vehicle 1	97.8	106.0	7.0	69.9	-1.0	0.2	0.0	330.0	9320459
Vehicle 2	2.3	1.0	0.0	2.7	1.4	1.3	0.0	18.0	223624
Vehicle 3	0.9	1.0	0.0	1.0	1.3	1.2	0.0	9.0	87232
Vehicle 4	1.0	1.0	0.0	1.3	3.5	1.7	0.0	12.0	96525
Vehicle 5	1.4	1.0	0.0	1.6	2.4	1.4	0.0	14.0	132275
Vehicle 6	1.6	1.0	0.0	1.9	1.7	1.4	0.0	15.0	151041
Vehicle 7	0.5	0.0	0.0	0.8	5.5	2.1	0.0	8.0	43926
Rain	0.0	0.0	0.0	0.1	1094.4	24.8	0.0	6.5	1238
Snow	4.3	0.0	-1.0	9.0	0.7	1.5	0.0	30.0	-
Temperature	7.8	6.6	0.5	10.1	-0.8	0.0	-20.3	31.6	-
Visibility	32929.9	37890.0	50000.0	16570.2	-1.2	-0.5	140.0	50000.0	-
Wind speed	4.3	4.0	3.5	2.0	0.5	0.7	0.0	14.7	-

Correlations between different variables are expressed in Appendix 3. As can be seen from the appendix, the speed lags have the expected high positive correlation with the speed variable itself. Thus, the count variables correlate negatively with the speed variable which is anticipated. Afternoon variable seems also correlating negatively with the speed as Plot 6 indicates.

4.7 TMS point 107

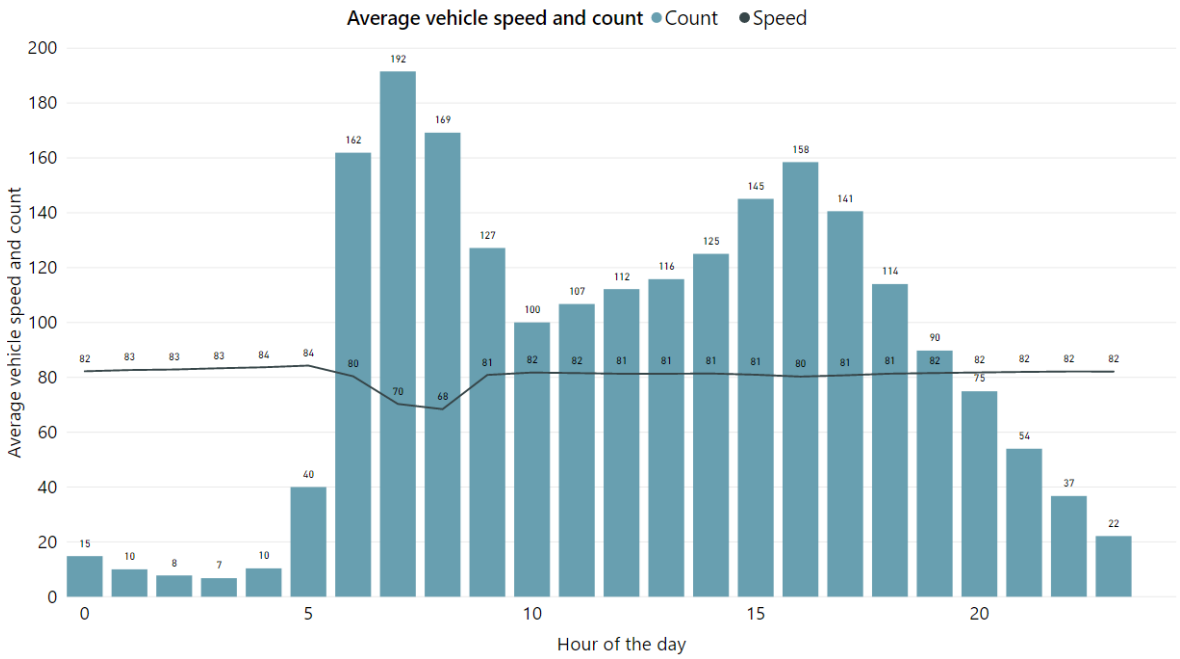
Plot 7 visualizes the change in speed and count during the observation period at point 107. Like in TMS 149, there are spikes in the speed and count during the time period. Although the variation in speed exists compared to TMS 149, the speed seems to be more stable during the time period. The maximum recorded speed that occurred is 118.5 km/h and the minimum is 13 km/h. The standard deviation (7.2 km/h) of the speed during the period is rather close to the first point. The largest recorded count was 371 vehicles during the 5-minute period and the standard

deviation for the count is 72. Previous TMS point for the point recorded the second largest count of the analysis, totally 434 vehicles.



Plot 7. TMS 107 speed and count

The average speed of the hour stays relatively the same the whole time. The mean speed in the dataset is 80.4 km/h and the median speed is 81.9 km/h. A small drop in average speed occurs between 7 and 9 because of the morning traffic (Plot 8). In addition, the largest number of traffic occurs during the speed drops like in the first point. There is a slight increase in the average vehicle amount during the afternoon, but it seems not to affect the speed on average.



Plot 8. The average speed and count by the hour of the day at point 107

Table 8. TMS 107 variable statistics

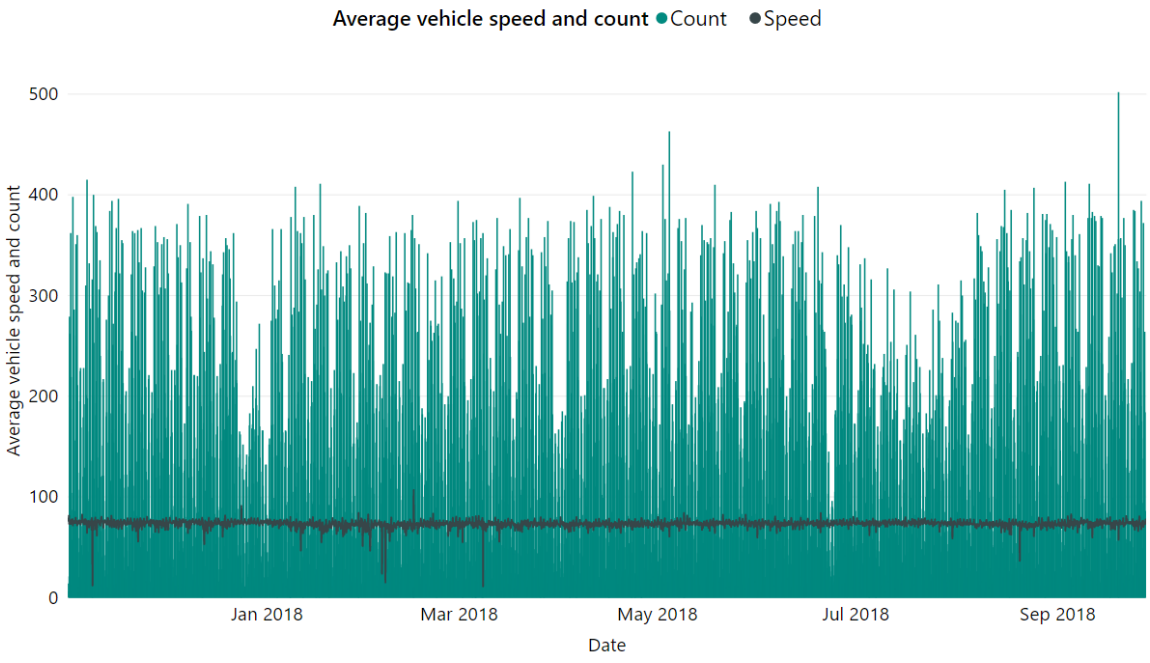
Variable	Mean	Median	Mode	Standard Deviation	Kurtosis	Skewness	Min	Max	Sum
Speed	80.8	81.9	83.0	7.2	25.5	-4.6	13.0	118.5	7700825
Speed previous TMS point	88.8	89.4	89.0	5.3	56.6	-5.9	16.1	125.0	8462550
Count	89.0	86.0	7.0	72.7	0.7	0.9	1.0	371.0	8479569
Count previous TMS point	98.2	97.0	7.0	80.0	1.1	1.0	1.0	434.0	9357030
Vehicle 1	82.3	80.0	5.0	67.9	0.8	0.9	0.0	353.0	7847304
Vehicle 2	3.1	1.0	0.0	3.8	1.9	1.5	0.0	29.0	298544
Vehicle 3	0.8	0.0	0.0	1.1	2.7	1.6	0.0	9.0	80915
Vehicle 4	1.2	1.0	0.0	1.4	2.1	1.4	0.0	12.0	111559
Vehicle 5	0.8	0.0	0.0	1.2	5.5	2.1	0.0	13.0	71716
Vehicle 6	0.6	0.0	0.0	0.9	3.6	1.8	0.0	8.0	53400
Vehicle 7	0.2	0.0	0.0	0.4	11.1	3.0	0.0	6.0	16108
Rain	0.0	0.0	0.0	0.1	1094.4	24.8	0.0	6.5	1238
Snow	4.3	0.0	-1.0	9.0	0.7	1.5	0.0	30.0	-
Temperature	7.8	6.6	0.5	10.1	-0.8	0.0	-20.3	31.6	-
Visibility	32929.9	37890.0	50000.0	16570.2	-1.2	-0.5	140.0	50000.0	-
Wind speed	4.3	4.0	3.5	2.0	0.5	0.7	0.0	14.7	-

Most of the traffic at point 107 are regular cars or vans which can be seen from the total amount of Vehicle 1 category observations. The second largest group is Vehicle 2 with almost 300 thousand records and the smallest group is Vehicle 7 with

slightly over 16 thousand. The selected weather station is the same for all three measurement points so there are no changes in the weather data compared to the first.

4.8 TMS point 126

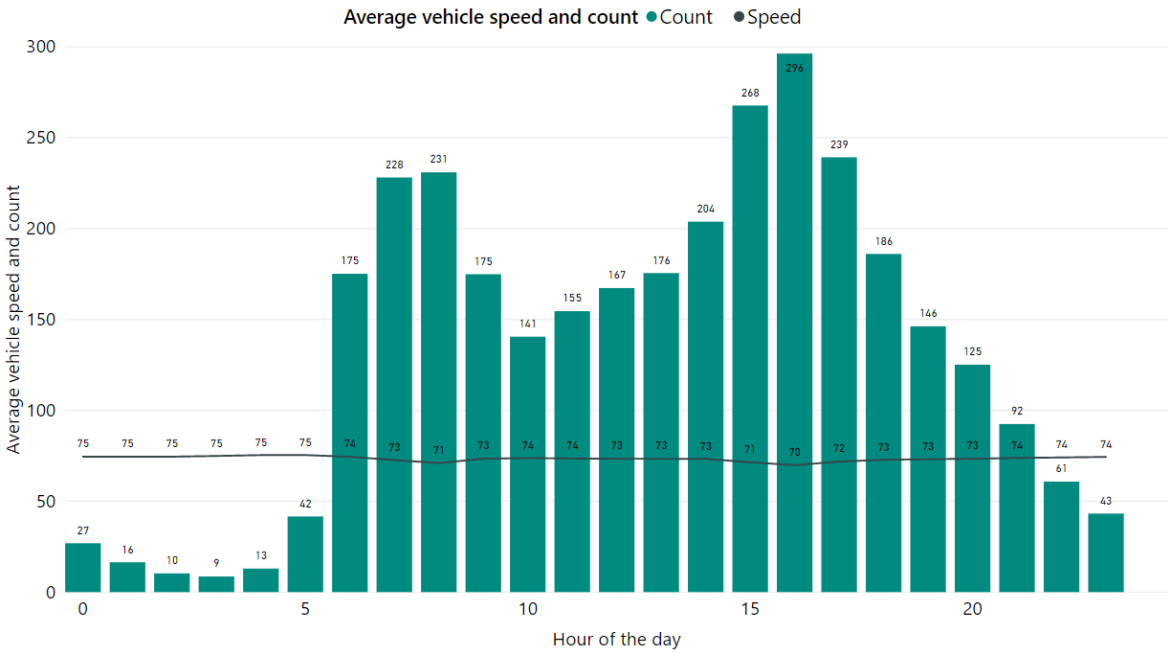
Comparing to the other two points, TMS 126 has the most constant variation in the average speed. The standard deviation during the time period is 3.2 km/h which indicates that there is no clear decrease in speed during the morning or afternoon traffic. Few spikes can be found from the dataset and the smallest recorded speed is 11.2 km/h. The maximum speed instead is 106.8 km/h which is the only distinct speed increase in the dataset. Minimum and maximum speed from the previous TMS point are around the same (14.3 km/h and 104 km/h) as in 126.



Plot 9. TMS 126 speed and count

The stability of the speed is also clear from the average speed plot by the hour of the day (Plot 10). The lowest average speed 70 km/h takes place during the 16 which does not differ substantially from the average speed of the dataset (73.5 km/h). Traffic flow increases significantly during the morning hours and even more

during the afternoon hours, but it does not have a notable effect on the average speed overall.



Plot 10. The average speed and count by the hour of the day at point 126

Table 9. TMS 126 variable statistics

Variable	Mean	Median	Mode	Standard	Kurtosis	Skewness	Min	Max	Sum
Speed	*73.5*	*73.7*	*74.0*	*3.2*	*51.2*	*-4.0*	*11.2*	*106.8*	*7001238*
Speed previous TMS point	*76.0*	*77.1*	*78.0*	*5.3*	*38.5*	*-5.5*	*14.3*	*104.0*	*7243158*
Count	*134.4*	*132.0*	*9.0*	*105.1*	*-0.5*	*0.6*	*1.0*	*502.0*	*12809617*
Count previous TMS point	*155.2*	*155.0*	*11.0*	*118.3*	*-0.6*	*0.5*	*1.0*	*531.0*	*14790157*
Vehicle 1	*127.2*	*125.0*	*6.0*	*100.6*	*-0.4*	*0.6*	*0.0*	*483.0*	*12122996*
Vehicle 2	*3.0*	*1.0*	*0.0*	*3.7*	*2.2*	*1.5*	*0.0*	*29.0*	*287884*
Vehicle 3	*0.4*	*0.0*	*0.0*	*0.7*	*5.4*	*2.0*	*0.0*	*10.0*	*40089*
Vehicle 4	*1.4*	*1.0*	*0.0*	*1.7*	*2.1*	*1.4*	*0.0*	*13.0*	*136927*
Vehicle 5	*1.2*	*1.0*	*0.0*	*1.4*	*2.7*	*1.5*	*0.0*	*13.0*	*109634*
Vehicle 6	*0.9*	*0.0*	*0.0*	*1.2*	*2.6*	*1.6*	*0.0*	*10.0*	*85240*
Vehicle 7	*0.3*	*0.0*	*0.0*	*0.6*	*7.4*	*2.5*	*0.0*	*6.0*	*26847*
Rain	*0.0*	*0.0*	*0.0*	*0.1*	*1094.4*	*24.8*	*0.0*	*6.5*	*1238*
Snow	*4.3*	*0.0*	*-1.0*	*9.0*	*0.7*	*1.5*	*-1.0*	*30.0*	*409538*
Temperature	*7.8*	*6.6*	*0.5*	*10.1*	*-0.8*	*0.0*	*-20.3*	*31.6*	-
Visibility	*32929.9*	*37890.0*	*50000.0*	*16570.2*	*-1.2*	*-0.5*	*140.0*	*50000.0*	-
Wind speed	*4.3*	*4.0*	*3.5*	*2.0*	*0.5*	*0.7*	*0.0*	*14.7*	-

The average count in the last TMS point is clearly the largest (134.4) comparing the other two points. The largest amount of traffic that was recorded is 502 vehicles during the 5-minute period which is explicitly the largest recorded amount of all the studied datasets. A larger amount of traffic also leads to a higher standard deviation which is 105.1. Like in the other two points, most of the count variable observations are composed of Vehicle 1- type vehicles. There are totally over 12 million cars during the observation period. The second largest group is Vehicle 2 with almost 300 thousand records and the smallest group is Vehicle 7 with slightly less than 30 thousand. The is no change in the weather statistics.

5. MODEL SELECTION FOR SPEED PREDICTING

In order to do the comparison of different models, all three datasets are split into a training set and test sets. The purpose of the training sets is to train the models and find the optimal order, coefficients or parameters. After the training, model performance is validated with a test set using selected performance measures. Datasets are split with 70/30 relation where 70 percent of the data is used for training and the remaining 30 percent is held out for validation. The training of the time series models is done in a traditional way using the first 70 percent of the data to find the order, whereas, for the other models, 5-fold cross-validation is applied for training. Before training and testing, all explanatory variables are normalized using min-max normalization.

5.1 Time Series

Before conducting the time series models, the stationarity condition of the speed is tested. For testing the condition, the Augmented Dickey-Fuller test and Ljung-Box test are performed to state the stationarity condition. ADF test shows the p-value to be $p = 0.01$ which indicates the lack of unit root and the data is stationarity. Ljung-Box test in turn states the p-value to be $p < 2.2e^{-16}$ which also indicates that the stationarity condition is fulfilled. P-values for TMS 107 and 126 are also $p < 2.2e^{-16}$ which justifies the stationary condition for the speed in every measurement point. To understand the speed lag and moving average of all three time series, the autocorrelation and partial autocorrelation functions are plotted in Appendix 4. Based on the plots, all three time series have geometrically decaying ACF. Meanwhile, the partial autocorrelation function indicates significant PACF coefficients until lag 18 for all time series.

To get better insight selecting the order of ARIMA, the information criterion is also used. Minimizing AIC and BIC values and using the maximum lag order to be 12 (past hour) the optimized model for 149 is ARIMA(1,0,4) which corresponds to the ARMA(1,4) model. The AIC value for the ARMA(1,4) is 335664.3 and the BIC value

is 335728.1. Based on the AIC the optimal model for 107 is ARIMA(5,1,0) and with BIC ARIMA(0,1,1). Between these two, ARIMA(5,1,0) is selected for further analysis. For point 126, the optimized order according to AIC and BIC is ARIMA(0,1,5) which is then selected to represent the second time series model in the TMS 126 analysis. Residual autocorrelation is also checked for every time series model and there is no significant autocorrelation in residuals. Coefficients of each time series model are expressed in tables 10 and 11.

Table 10. AR(1) coefficients

AR	AR1	Log likelihood	AIC	BIC	p-value	RMSE
TMS 149	*0.9114*	*-171050.6*	*342107*	*342134.6*	*2.2e-16*	*3.1417*
TMS 107	*0.8976*	*-177339.6*	*354685*	*354712.6*	*2.2e-16*	*3.4522*
TMS 126	*0.7127*	*-151052.3*	*302111*	*302138*	*2.2e-16*	*2.328*

Table 11. ARIMA coefficients

ARIMA(1,0,4)	AR1	MA1	MA2	MA3	MA4	Log likelihood	AIC	BIC	RMSE
TMS 149	*0.9579*	*-0.3388*	*0.12*	*0.0158*	*0.0332*	*-167825.1*	*335664.3*	*335728.1*	*2.9934*
p-value	*<2.2e-16*	*<2.2e-16*	*0.00399*	*0.00001*	*<2.2e-16*				
ARIMA(5,1,0)	AR1	AR2	AR3	AR4	AR5	Log likelihood	AIC	BIC	RMSE
TMS 107	*-0.3389*	*-0.1116*	*-0.0409*	*-0.0118*	*-0.0096*	*-175460.8*	*350933.5*	*350988.2*	*3.3565*
p-value	*<2.2e-16*	*<2.2e-16*	*<2.2e-16*	*0.003805*	*0.012911*				
ARIMA(0,1,5)	MA1	MA2	MA3	MA4	MA5	Log likelihood	AIC	BIC	RMSE
TMS 126	*-0.6009*	*-0.0317*	*-0.0104*	*-0.0018*	*-0.049*	*-145957.6*	*291927.2*	*291981.9*	*2.1569*
p-value	*<2.2e-16*	*2.2e-16*	*0.02437*	*0.68977*	*<2.2e-16*				

5.2 Linear Regression

There are totally 26 explanatory variables that are used to conduct linear regression. In Table 12 are the variable coefficients from the training set. Based on the results, all speed lags are significant with a $p < 0.001$. Other equally significant variables are previous TMS point count lag, count lag 2, count lag 3, a minute of the hour, morning time, afternoon time and from the weather variables visibility and wind speed. The residual standard error is 2.938 on 66692 degrees of freedom. R-squared is 0.8501 and the adjusted R-squared is 0.8501. The F-statistic is

1.455e+04 on 26 and 66692 degrees of freedom and the P-value is $p < 2.2e^{-16}$. 10- and 15-minute training result coefficients can be found from Appendix 5. Root mean squared error and mean absolute error for the trained model are 2.94 and 1.93, respectively.

Table 12. TMS 149 5-minute regression analysis coefficients

Point 149	Estimate	Std. Error	t value	Pr (>|t|)	
Intercept	26.5193	0.2845	93.2142	0.0000	***
Speed lag 1	44.8869	0.3212	139.7304	0.0000	***
Speed lag 2	16.3595	0.3574	45.7742	0.0000	***
Speed lag 3	5.8324	0.3124	18.6724	0.0000	***
Speed lag prev. point	9.3328	0.2595	35.9613	0.0000	***
Count lag 1	58.3499	76.9509	0.7583	0.4483	
Count lag 2	0.9281	0.2681	3.4611	0.0005	***
Count lag 3	-0.8061	0.2406	-3.3506	0.0008	***
Count lag prev. point	-13.0136	0.2641	-49.2674	0.0000	***
Vehicle 1	-47.3394	76.0291	-0.6226	0.5335	
Vehicle 2	-2.3152	4.1478	-0.5582	0.5767	
Vehicle 3	-0.0765	2.0755	-0.0369	0.9706	
Vehicle 4	-1.6112	2.7663	-0.5824	0.5603	
Vehicle 5	-1.4023	3.2271	-0.4346	0.6639	
Vehicle 6	-1.4564	3.4581	-0.4212	0.6736	
Vehicle 7	-1.2038	1.8475	-0.6516	0.5147	
Hour	-0.0244	0.0465	-0.5245	0.6000	
Minute	0.2806	0.0364	7.7070	0.0000	***
Afternoon time	-0.7081	0.0480	-14.7503	0.0000	***
Morning time	0.4052	0.0388	10.4322	0.0000	***
Weekend	0.1706	0.0291	5.8602	0.0000	***
Holiday	0.1070	0.0526	2.0361	0.0417	*
Rain	-1.6562	0.8676	-1.9090	0.0563	.
Snow	0.0292	0.0542	0.5384	0.5903	
Temperature	-0.0965	0.0838	-1.1520	0.2493	
Visibility	0.1505	0.0368	4.0936	0.0000	***
Wind speed	-0.2935	0.0852	-3.4439	0.0006	***

Significance codes: p<0 ***, p<0.001 **, p<0.01 *, p<0.1 .

Next, the regression analysis is operated to point 107. The 5-minute coefficients for the explanatory variables are in Table 13 and there is no major change comparing the first TMS point. All the speed lags achieve the $p < 0.001$ significance level. From the time or day-related variables, almost every single one is also significant in a highest level. Like in the first TMS point, none of the vehicle classes is significant and from the weather aspect, $p < 0.001$ is fulfilled only by temperature. The residual

standard error is 3.228 on 66688 degrees of freedom. Multiple R-squared of the model is 0.8516 and the adjusted R-squared is 0.8515. The F-statistic is $1.472e^4$ on 26 and 66688 degrees of freedom and the P-value is $p < 2.2e^{-16}$. Root mean squared error and mean absolute error for the trained model are 3.23 and 1.97. Variable coefficients for 10- and 15-minute time periods can be found from the Appendix 5.

Table 13. TMS 107 5-minute regression analysis coefficients

Point 107	Estimate	Std. Error	t value	Pr(>|t|)	
Intercept	16.89999274	0.69758678	24.2263661	4.26E-129	***
Speed lag 1	63.26370267	0.41348935	152.9995923	0.00E+00	***
Speed lag 2	18.08713147	0.47396055	38.161681	3.116159E-315	***
Speed lag 3	5.26945479	0.40529419	13.0015551	1.34E-38	***
Speed lag prev. point	12.81865952	0.29406769	43.5908466	0.00E+00	***
Count lag 1	136.7951908	249.197046	0.5489439	5.83E-01	
Count lag 2	0.60681647	0.36558859	1.6598343	9.70E-02	.
Count lag 3	2.19996438	0.31497677	6.9845289	2.88E-12	***
Count lag prev. point	-12.32414743	0.30279908	-40.7007432	0.00E+00	***
Vehicle 1	-126.7481491	237.7484252	-0.5331188	5.94E-01	
Vehicle 2	-9.48930815	19.53174428	-0.4858403	6.27E-01	
Vehicle 3	-2.70492921	6.06234536	-0.4461853	6.55E-01	
Vehicle 4	-3.00394901	8.08240423	-0.3716653	7.10E-01	
Vehicle 5	-3.81514351	8.75678442	-0.4356786	6.63E-01	
Vehicle 6	-2.04502141	5.3895212	-0.379444	7.04E-01	
Vehicle 7	-1.50321594	4.04481523	-0.3716402	7.10E-01	
Hour	0.23192488	0.04941778	4.6931467	2.70E-06	***
Minute	0.28310492	0.04002039	7.0740172	1.52E-12	***
Afternoon time	0.40848983	0.04383354	9.3191167	1.21E-20	***
Morning time	-0.13723624	0.04679236	-2.9328773	3.36E-03	**
Weekend	0.48353385	0.03397488	14.2321006	6.76E-46	***
Holiday	0.20568982	0.05984102	3.4372715	5.88E-04	***
Rain	-1.15778405	0.95014938	-1.2185284	2.23E-01	
Snow	0.04204637	0.05943397	0.7074467	4.79E-01	
Temperature	0.44097637	0.09100805	4.8454653	1.27E-06	***
Visibility	0.08971385	0.04062874	2.2081374	2.72E-02	*
Wind speed	0.1499163	0.09271223	1.6170067	1.06E-01	

Significance codes: p<0 ***, p<0.001 **, p<0.01 *, p<0.1 .

The last TMS point differs from the first two because almost every weather variable is significant when the $p < 0.001$. Only wind speed does not have a significant coefficient based on the model. Speed lags continue to be significant also in point 107 training set such as count lag from previous TMS point, count lag 3, minute and

afternoon time. Morning time and weekend variables achieve the significance level of $p = 0.05$. The residual standard error is 2.062 on 66692 degrees of freedom. Multiple R-squared is 0.5709 and the adjusted R-squared is 0.5707. The F-statistic is 3412 on 26 and 66692 degrees of freedom and P-value is $p < 2.2e^{-16}$. Root mean squared error and mean absolute error for the trained model are 2.06 and 1.35. 10- and 15-minute training set coefficients can be found from Appendix 5.

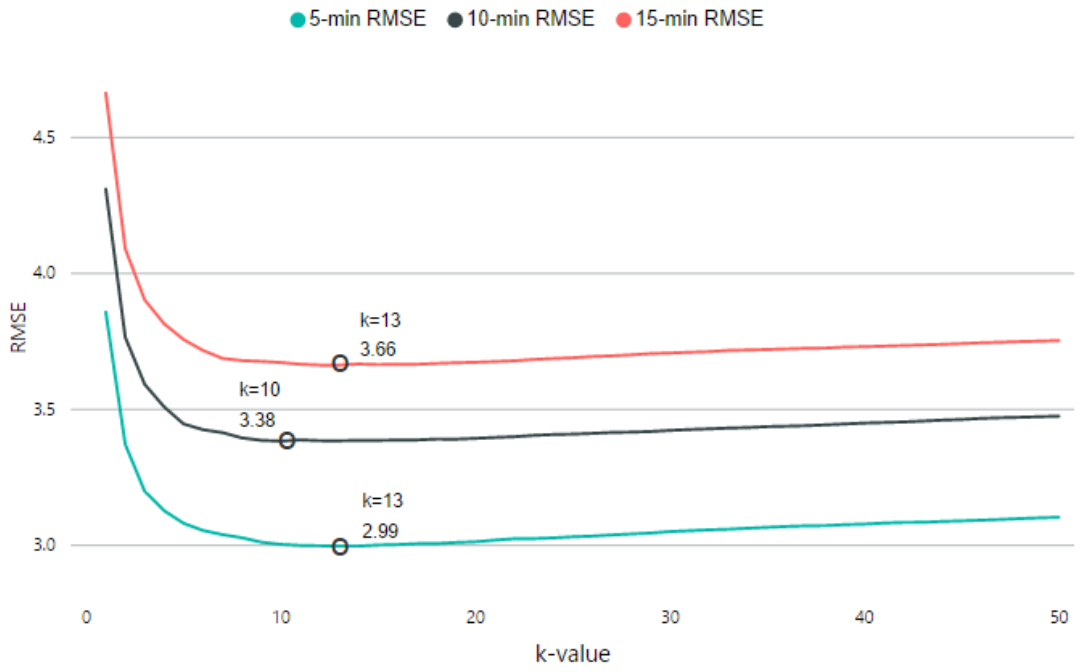
Table 14. TMS 126 5-minute regression analysis coefficients

Point 126	Estimate	Std. Error	t value	Pr(>|t|)	
Intercept	27.7836	0.7661	36.2685	0.0000	***
Speed lag 1	33.8542	0.3736	90.6249	0.0000	***
Speed lag 2	19.5289	0.3799	51.4033	0.0000	***
Speed lag 3	13.7977	0.3638	37.9255	2.067744e-311	***
Speed lag prev. point	5.0218	0.1680	29.8930	0.0000	***
Count lag 1	536.8066	365.2171	1.4698	0.1416	
Count lag 2	0.1168	0.2476	0.4718	0.6371	
Count lag 3	0.8225	0.2120	3.8804	0.0001	***
Count lag prev. point	-5.4015	0.2370	-22.7912	0.0000	***
Vehicle 1	-514.6394	352.0956	-1.4616	0.1438	
Vehicle 2	-30.3543	21.1405	-1.4358	0.1511	
Vehicle 3	-10.7804	7.2907	-1.4786	0.1392	
Vehicle 4	-13.5843	9.4769	-1.4334	0.1517	
Vehicle 5	-13.6067	9.4768	-1.4358	0.1511	
Vehicle 6	-10.2463	7.2902	-1.4055	0.1599	
Vehicle 7	-6.2323	4.3747	-1.4246	0.1543	
Hour	0.0033	0.0317	0.1057	0.9158	
Minute	0.1413	0.0255	5.5318	0.0000	***
Afternoon time	-0.2913	0.0325	-8.9668	0.0000	***
Morning time	0.0692	0.0290	2.3896	0.0169	*
Weekend	0.0506	0.0208	2.4319	0.0150	*
Holiday	0.0567	0.0370	1.5354	0.1247	
Rain	-3.8795	0.6116	-6.3432	0.0000	***
Snow	-0.4845	0.0388	-12.4880	0.0000	***
Temperature	-0.2968	0.0580	-5.1185	0.0000	***
Visibility	0.1092	0.0258	4.2332	0.0000	***
Wind speed	-0.0004	0.0596	-0.0064	0.9949	

Significance codes: p<0 ***, p<0.001 **, p<0.01 *, p<0.1 .

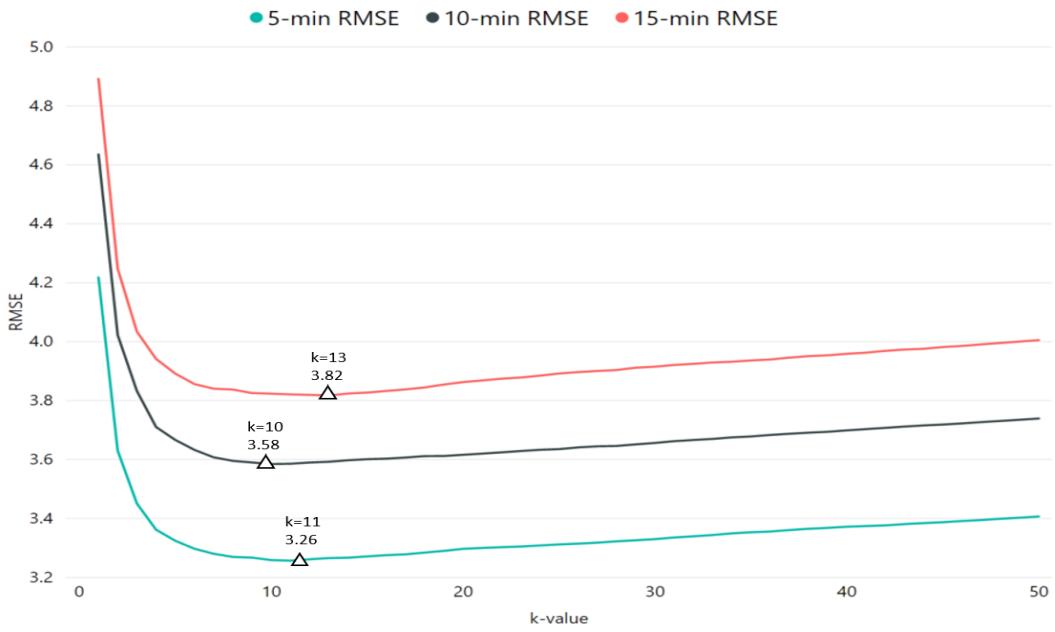
5.3 K-Nearest Neighbor

For deciding the optimal K-value for the KNN model based on the training set, the RMSE value is minimized testing K = 1 to K = 50. The 5-minute prediction the optimal K-value settles to K=13 as in the 15-minute prediction. In 10-minute forecasting, the optimal K-value is 10. The RMSE values with different time predictions are visualized using the elbow method in Plot 11. The blue line represents the 5-minute RMSE value, black line, in turn, is the 10-minute RMSE value and the red one indicates the 15-minute values. Sample sizes for the 5-, 10- and 15-minute periods are 66719, 66718 and 66717.

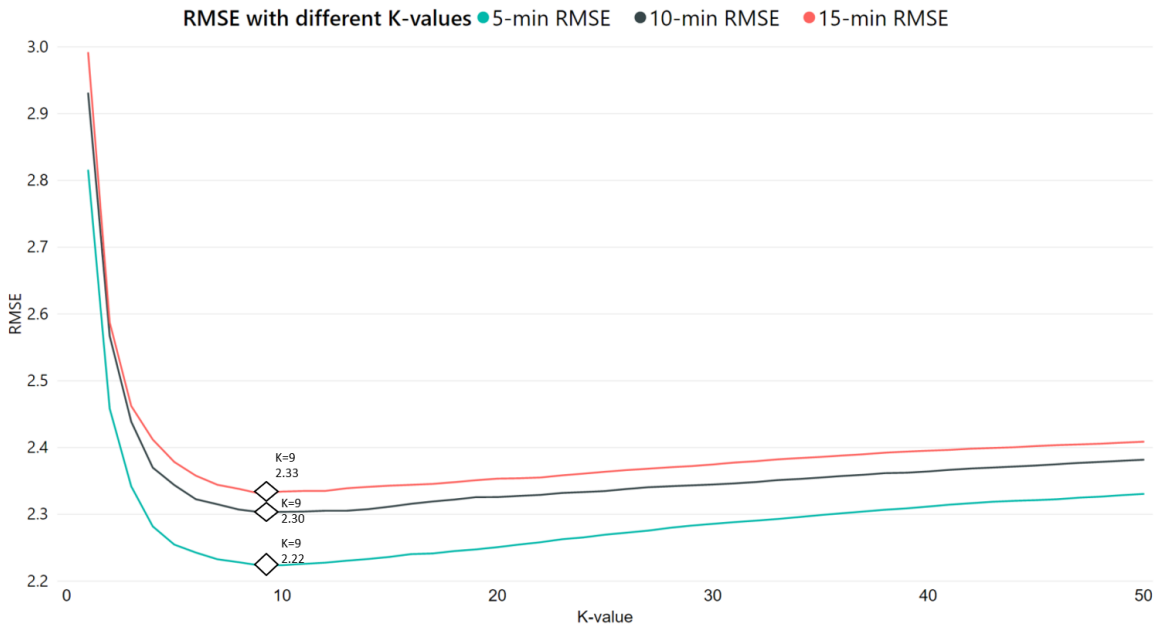


Plot 11. Point 149 K-value with the smallest RMSE value

In point 107 the optimal K-values are decided using the same method as in 149. For the 5-minute estimation period, the minimum RMSE value (3.26) was achieved using K=11. For the 10-minute and 15-minute periods, the k-values with RMSE 3.58 and 3.82 are K=10 and K=13.



Plot 12. Point 107 K-value with the smallest RMSE value



Plot 13. Point 126 K-value with the smallest RMSE value

The last of three points optimal K-value settles to K = 9 in every time periods. Training set RMSE for 5-minute, 10-minute and 15-minute are 2.22, 2.30 and 2.33. The elbow visualization is expressed in Plot 13.

5.4 Extreme Gradient Boosting

Comparing all the other models in the thesis, the Extreme Gradient Boosting has various different possibilities to establish the final model. Construction is done by parameter optimization in order to achieve the best model according to the training set. Using parameter tuning superior models can be obtained but it may also lead to overfitting which may result as unsatisfactory predictions. For the XGBoost there are three types of parameters: General parameters, Booster parameters and task parameters. Following parameters were chosen for the grid search:

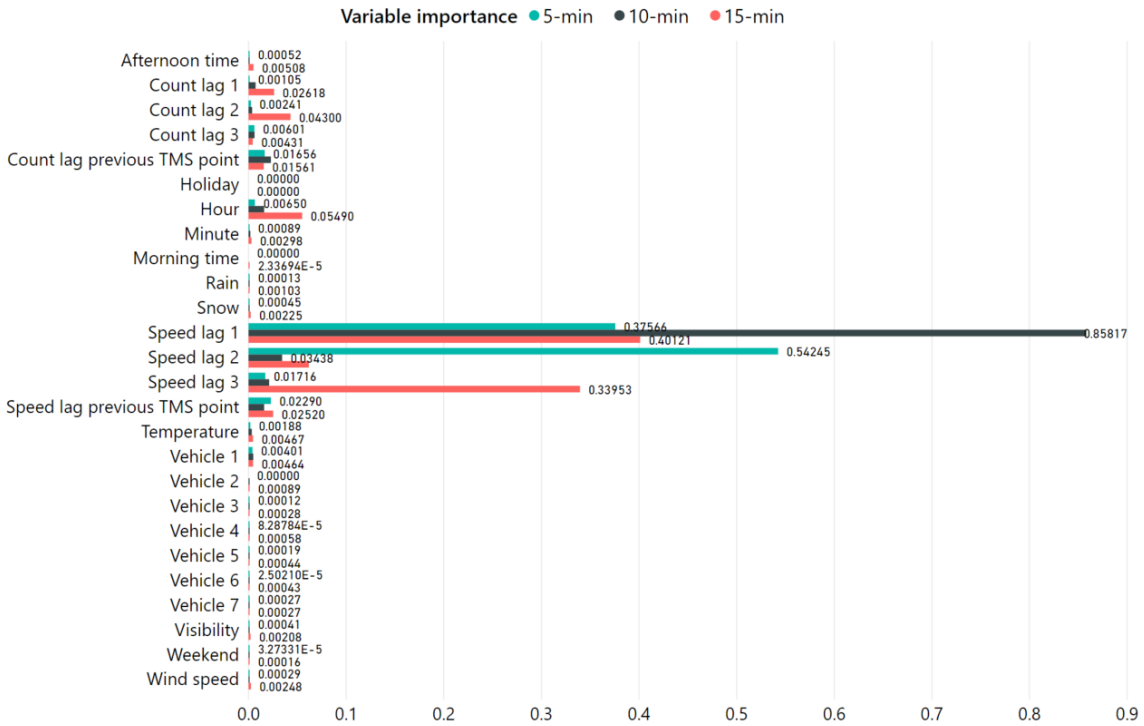
- ***Eta***, also known as ***Learning rate,*** shrinks each feature weights after boosting step to make the process simpler and makes the model less prone to overfitting.
- ***Gamma*** expresses the loss reduction at minimum that is needed to do the onwards partitioning on a leaf node of the decision tree.
- ***Max depth*** defines the maximum depth of a tree. The deeper the tree gets the more complex the model is which can lead to overfitting.
- ***Colsample bytree*** is a column subsample ratio when a tree is established.
- ***Subsample*** is ratio that tells how much of the data is utilized to grow trees.
- ***Nrouds*** is the maximum number of boosting iterations.
- ***Min child weight*** is the minimum sum of instance weight that is required in a child

Detailed list of all possible parameters and their meanings can be found from the XGBoost Documentation (2018). For simplicity, gamma and min child weight are held constant for all trained models. The following training parameters were achieved after cross-validation and grid search:

Table 15. XGBoost parameter values

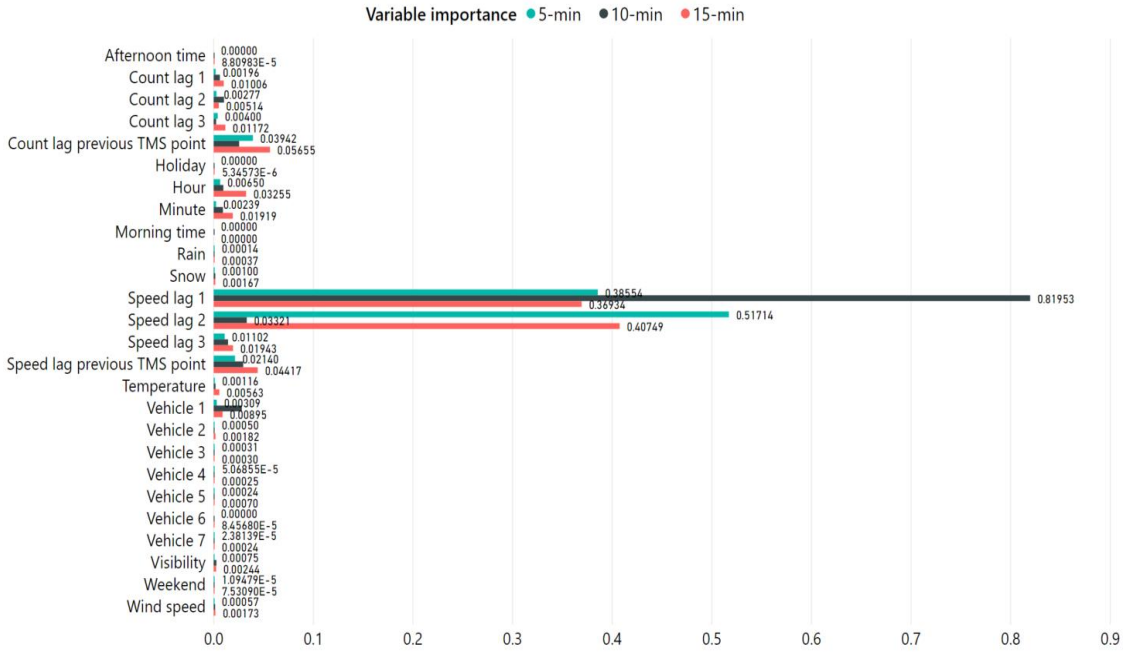
TMS	Time	Eta	Max depth	Colsample bytree	Subsample	Nrounds	Gamma	Min child weight	RMSE
149	*5*	*0.3*	*3*	*0.8*	*1*	*100*	*0*	*1*	*2.636717*
	10	*0.3*	*3*	*0.8*	*1*	*100*	*0*	*1*	*3.075462*
	15	*0.3*	*3*	*0.6*	*1*	*150*	*0*	*1*	*3.351009*
107	*5*	*0.3*	*3*	*0.8*	*1*	*100*	*0*	*1*	*2.897187*
	10	*0.3*	*3*	*0.8*	*1*	*150*	*0*	*1*	*3.297747*
	15	*0.3*	*3*	*0.8*	*1*	*150*	*0*	*1*	*3.548952*
126	*5*	*0.3*	*3*	*0.6*	*1*	*100*	*0*	*1*	*1.956348*
	10	*0.3*	*3*	*0.8*	*1*	*150*	*0*	*1*	*2.122396*
	15	*0.3*	*3*	*0.6*	*1*	*150*	*0*	*1*	*2.168921*

The grid search examples at TMS 149 with a 5-minute time frame can be found from Appendix 6. In Plot 14 can be seen the importance of the variables with different time frames. The measure expresses the improvement considering the accuracy that each feature brought to the branch. Variable importance from TMS 107 and 126 are found from Plot 15 and Plot 16.

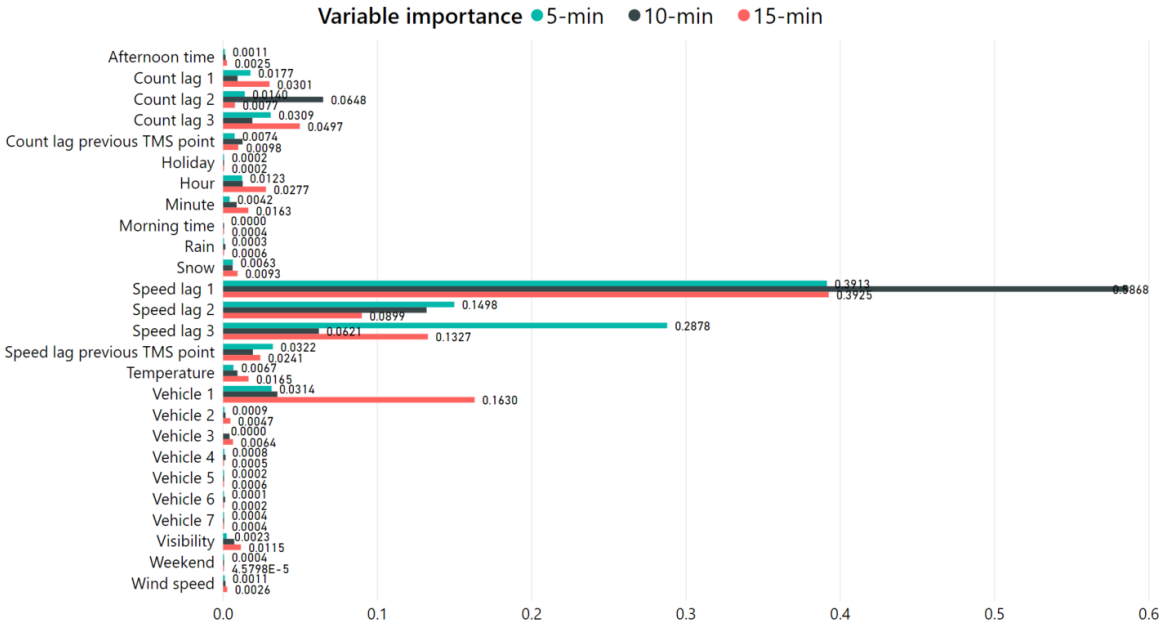


Plot 14. TMS 149 XGBoost variable importance

As expected, the speed lags contribute most of the model accuracy within every time frame. Surprisingly, speed lag 2 was the most important factor during the 5-minute period in the first two points. In other cases, speed lag 1 brings the greatest contribution for each model. Another unexpected fact is that the count variables do not add much to the models. Other variables seem not to add anything important to the final models.



Plot 15. TMS 107 XGBoost variable importance



Plot 16. TMS 126 XGBoost variable importance

Other two points show almost the same results what comes to the variable importance. Speed lag 3 at TMS 107 seems not to contribute much with the predicted speed, whereas, in 126 the significance is far more explicit. In point 126 the average speed stays more stable than in the other points. That increases the importance of last 5-minute speed compared to other ones. A clear change in 126 comparing the other two is that vehicle class 1 seems to have a reasonable effect on the model during the 15-minute estimation period. Thus, the effect of other vehicle classes is negligible within every time period.

6. RESULTS

In this chapter, the results of the selected models are presented and reported. For the speed predicting perspective, RMSE and MAPE values are obtained from each model. The predicting time frames are 5, 10 and 15 minutes forward. Model performance during the speed drops is also included in this section, and the chapter also provides results for the jam classification problem.

6.1 TMS 149

RMSE and MAPE values for different models at TMS point 149 are expressed in Table 16 (LM refers to Linear Regression). The values explain which model performed the best based on the evaluation metric. The 5-minute prediction period provides the best results overall with every model which is expected. Similar findings were also achieved by Yu et al. (2018). Increasing the forecasting interval from the present to the future increases the uncertainty, which results as more imprecise predictions. The most accurate is the Extreme Gradient Booster in both categories. RMSE value for the XGBoost is 2.549 and MAPE 2.465. Clearly, the worst performing model during the 5-minute period in both categories is the AR(1) model with RMSE value 3.277 and MAPE value 3.284. Linear regression and KNN also produced decent results and are rather close to each other. ARIMA model, on the other hand, was the second worst model for all periods.

For the 10- and 15-minute periods there is no significant change in the order of the model performance. XGBoost provides the most accurate results for the 10-minute period: RMSE value 2.972 and MAPE value 2.747. For the 15-minute period, the same values are 3.243 and 2.902. The second accurate model in both time frames is the KNN model and the third best one is the Linear model. The worst model is AR(1) as during the 5-minute period with RMSE values 3.751 and 4.190 and MAPE values 3.615 and 3.950.

Table 16. TMS point 149 RMSE and MAPE values with different periods

	5 min		*10 min*		*15 min*	
	RMSE	*MAPE*	*RMSE*	*MAPE*	*RMSE*	*MAPE*
AR(1)	*3.27786*	*3.28433*	*3.75103*	*3.61541*	*4.19016*	*3.95014*
ARIMA(1,0,4)	*3.11188*	*3.01405*	*3.62363*	*3.30984*	*4.05888*	*3.59899*
LM	*2.88404*	*2.7916*	*3.35902*	*3.10726*	*3.74589*	*3.38415*
KNN	*2.92518*	*2.75708*	*3.26375*	*2.9677*	*3.52409*	*3.14029*
XGB	*2.54903*	*2.46514*	*2.97291*	*2.74713*	*3.24328*	*2.90293*

Plot 17. TMS point 149 XGBoost actual and predicted values for the 5-minute period

Plot 17 above expresses the relationship between the predicted values of the XGBoost and actual values. The X-axis indicates the predicted values by the model whereas the Y-axis represents the actual speed. As the plot denotes, there are two clusters in the test set. Majority of the observations exists around the speed limit of the road, but as the lower cluster indicates, there are also a considerable amount of low-speed observations.

6.1.1 Time drop comparison

The second examined feature that is studied is the performance of the models during the speed drops. The goal is to see whether the models are able to predict future speed when the speed drop occurs. There are five different speed drop categories selected to perform the speed drop comparison: Speed drop smaller than 2 km/h, between 3-5 km/h, 7-10 km/h, 12-15 km/h and equal or greater than 20 km/h. The RMSE and MAPE values for different models with different speed drop categories can be seen from Table 17.

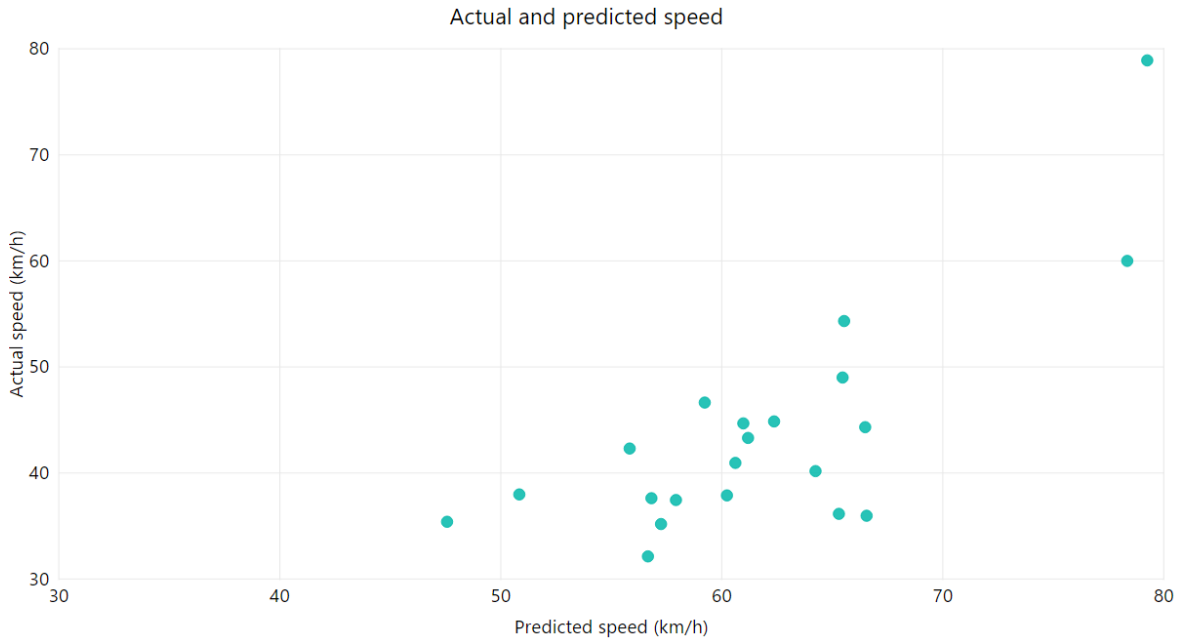
When the speed after the drop is close to the last measured speed value, a model that relies more on the previous speed values seem to give the best results. As it can be stated from Table 17 the time series models outperform other more complicated models when the speed drop is 2 km/h or less. The best performing model measured by RMSE (1.195) is the simple AR(1) model that uses only the first lag which is, in this case, the last 5-minute value. Using MAPE measure the best model is the ARIMA model with value 1.394. For the small speed changes the worst performing model is KNN model (RMSE: 1.792 and MAPE: 1.812)

When the speed drop increases, more complex models that use more dimensionalities for prediction are beginning to outperform simple time series models. When the drop is between 3-5 km/h and 7-10 km/h the prediction accuracies of the LM, KNN and XGB models are somewhat close to each other. Although, when the speed drop gets higher the XGBoost starts to outperform other models distinctly. When the drop increases higher than 20 km/h the XGBoost captures more variation than other models which indicates its abilities to work best if the large speed drop prediction is required. On the other hand, all the models under investigation seem to perform poorly during the sharp speed drops based on the error measures. This indicates that using the current dataset drops seem to be more or less noise or random which cannot be anticipated based on the previous data.

Table 17. TMS point 149 model performance with different speed drop levels

	< 2 km/h		3-5 km/h		7-10 km/h		12-15 km/h		>= 20 km/h	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
AR(1)	1.192	1.396	3.856	5.570	8.365	13.897	13.661	25.316	25.042	65.855
ARIMA(1,0,4)	1.261	1.387	3.473	4.840	7.571	12.323	13.329	24.578	25.812	68.203
LM	1.315	1.400	3.341	4.511	7.628	12.389	13.309	24.805	23.180	58.376
KNN	1.767	1.811	3.294	3.887	7.003	10.222	14.235	24.541	22.992	55.888
XGB	1.416	1.470	2.884	3.649	6.253	9.163	10.431	17.617	19.490	48.183

The predicted and actual speed of the XGBoost are visualized in Plot 18. As the plot denotes, the model seems to underestimate most of the drops compared to the actual speed drop. Based on the knowledge, the model does not capture speed drops when the decline is huge or abrupt.



Plot 18. TMS point 149 XGBoost drop >= 20 km/h actual and predicted values

6.2 TMS 107

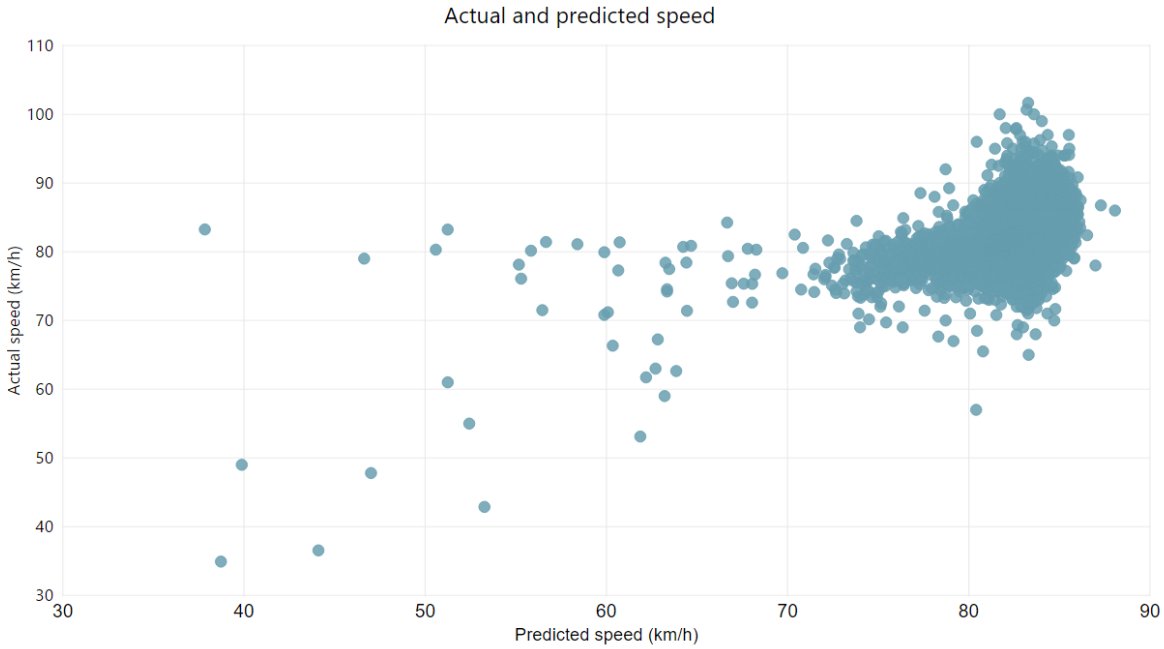
Overall, TMS 107 provides equivalent results comparing to the TMS 149. Although, the models seem to perform slightly better comparably to the TMS 149 root mean squared error and mean percentage error values. At the 5-minute prediction period,

the best model with RMSE 2.12 and MAPE 1.75 turns out to be XGBoost as well. The worst performing model during the same time period is the simple autoregressive model AR(1) with RMSE 2.90 and MAPE 2.39. KNN model performs slightly better during the 5-minute time frame than the linear model. For the 10-minute and 15-minute predictions, the order in model performance stays the same. Extreme Gradient Boosting still outperforms other models in both measures. On the other hand, the difference comparing to the KNN model during the 10-minute forecasting period is almost nonexistent. The worst performing model is still AR(1). However, the predicting ability using only the last average speed of vehicles seems to do rather well.

Table 18. TMS point 107 RMSE and MAPE values with different periods

	5 min		*10 min*		*15 min*	
	RMSE	*MAPE*	*RMSE*	*MAPE*	*RMSE*	*MAPE*
AR(1)	*2.908631*	*2.397432*	*3.218764*	*2.575757*	*3.465838*	*2.719228*
ARIMA(5,1,0)	*2.761125*	*2.216953*	*3.138271*	*2.386744*	*3.436062*	*2.514144*
LM	*2.541172*	*2.099784*	*2.673449*	*2.227332*	*2.838275*	*2.334863*
KNN	*2.275355*	*1.876852*	*2.232814*	*1.809606*	*2.425185*	*1.924659*
XGB	*2.122819*	*1.75435*	*2.227611*	*1.808972*	*2.26681*	*1.837153*

Plot 19 expresses the actual and predicted speed of the best performing model (XGBoost). Data points are from the 10-minute predictions and as it can be seen, data points settle around the speed limit of the road, apart from few exceptions in the test set. Data indicates that there are only a few situations where the speed is considerably lower which also explains better results considering the performance measures.



Plot 19. XGBoost 10-minute actual and predicted speed

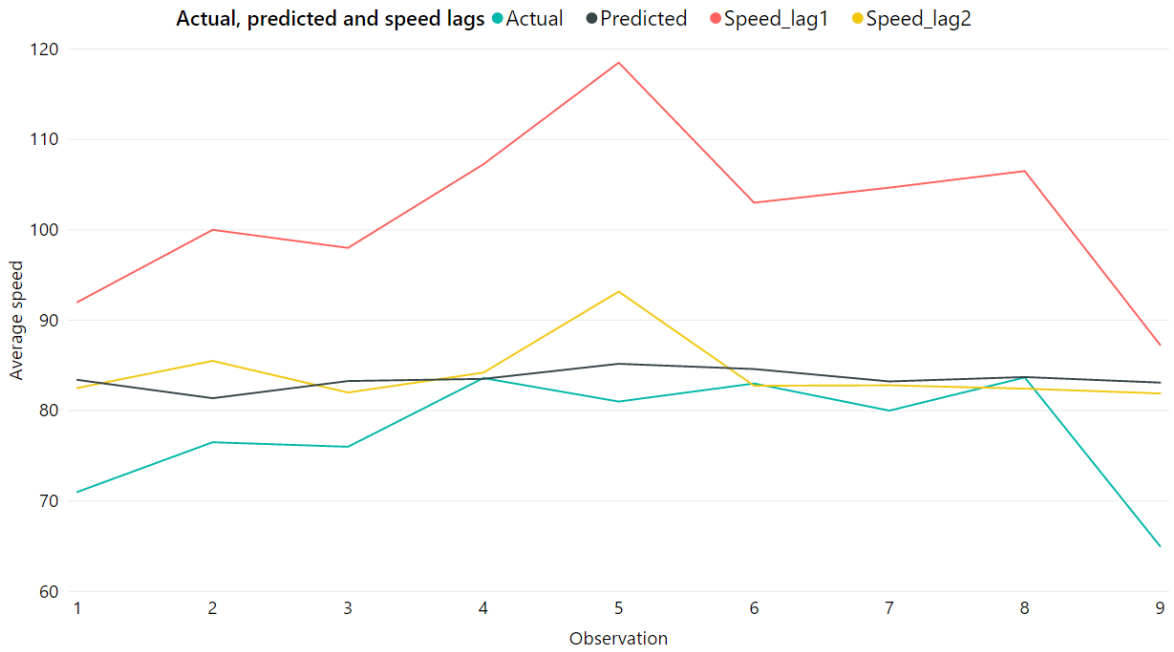
6.2.1 Speed drop comparison

Small speed changes are clearly controlled by the simple time series models also in TMS 107. The result seems to be rather obvious because almost all weight is given to the last occurred speed and when the speed does not change much, predictions are close to reality. AR(1) RMSE and MAPE values during the less than 2 km/h change are below one, but when it comes to the higher speed drop, the model starts to perform poorly. KNN is the worst model during the small drops which is the same finding than in TMS 149. When the speed drop increases, KNN and XGBoost start to surpass the time series models and linear model. For the 3-5 km/h speed drops, KNN and XGBoost perform the best. For the 7-10 km/h and 12-15 km/h drops the XGBoost is the best model while when the speed drop is larger than 20 km/h the KNN model beats the other models.

Table 19. 107 model performance with different speed drop levels

	< 2 km/h		3-5 km/h		7-10 km/h		12-15 km/h		>= 20 km/h	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
AR(1)	*0.915*	*0.894*	*3.606*	*4.595*	*7.966*	*11.210*	*13.138*	*21.029*	*24.385*	*47.026*
ARIMA(5,1,0)	*1.083*	*1.066*	*3.432*	*4.222*	*7.014*	*9.577*	*11.826*	*18.565*	*21.946*	*42.016*
LM	*1.148*	*1.122*	*3.386*	*3.978*	*6.703*	*8.291*	*10.302*	*13.223*	*17.615*	*22.408*
KNN	*1.450*	*1.324*	*2.847*	*2.872*	*4.934*	*5.400*	*7.081*	*7.873*	*8.087*	*8.059*
XGB	*1.240*	*1.139*	*2.788*	*2.952*	*4.877*	*5.380*	*6.973*	*7.699*	*8.568*	*8.784*

Comparing the RMSE and MAPE values in the case of XGBoost and KNN between point 149 and 107, there seems to be a significant improvement in predicting ability. When the speed drop is over 20km/h RMSE and MAPE using the KNN are 8.087 and 8.059 and using the XGBoost 8.568 and 8.784 compared to the point 149 values that are with KNN 22.992 and 55.888 and with XGBoost 19.49 and 48.183. In order to understand the reason behind the change in forecasting capability, it is reasonable to check the situations where the drop happened.



Plot 20. Actual, predicted and speed lag 1 of the KNN >=20 km/h drops

As an example, Plot 20 expresses the relationship between the actual and predicted speed of the KNN model during the 20 km/h drop or more. There are only 9 observations with the drop magnitude like that in the test set. Plot 20 indicates that

before the sharp drop, there has been an increase in speed. After the increase, the speed returns close to the average which causes the predictions to be closer to the actual speed. 12-15 km/h drops are also majorly caused by an increase in speed before the decrease which causes the models to perform at a satisfactory level. In TMS 149 almost all the speed drops occurred without the speed increase which leads to more worse RMSE and MAPE values. Linear and time series models react more clearly to the speed lag that results as more unsatisfactory results.

6.3 TMS 126

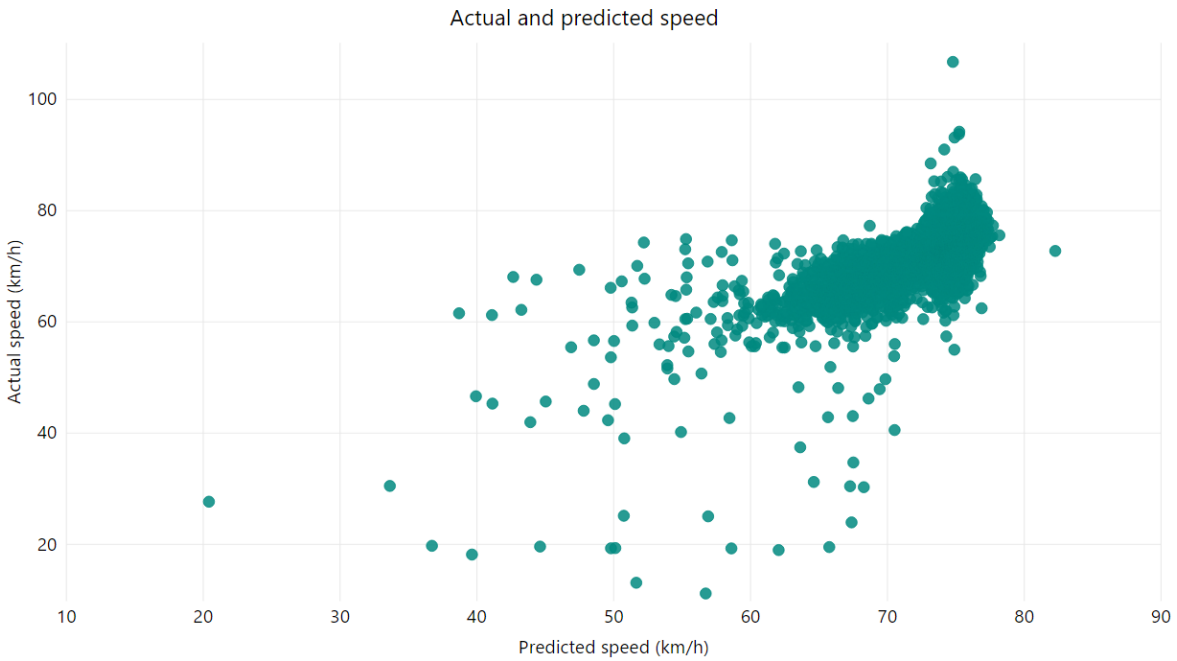
As Plot 9 revealed, TMS 126 has the most constant speed of all three observation points. Stability of the traffic speed can be also seen from the predictions of the models. Between 5-, 10- and 15-minute time frames, there is no substantial drop in predicting abilities in any model. Extreme Gradient Boosting turns out to be the most accurate model for all time periods. Surprisingly, the KNN model performs worse compared to more simple time series models. AR(1) holds the last place only when we are forecasting 15 minutes forward, otherwise, the time series models performances are satisfactory.

Table 20. TMS point 126 RMSE and MAPE values with different periods

	5 min		*10 min*		*15 min*	
	RMSE	*MAPE*	*RMSE*	*MAPE*	*RMSE*	*MAPE*
AR(1)	*2.15404*	*2.02375*	*2.25297*	*2.0734*	*2.36959*	*2.13921*
ARIMA(0,1,5)	*2.01342*	*1.85719*	*2.15173*	*1.93821*	*2.26606*	*2.00759*
LM	*2.07683*	*1.90029*	*2.17012*	*1.99539*	*2.27686*	*2.05958*
KNN	*2.26042*	*2.06434*	*2.27106*	*2.0964*	*2.27567*	*2.08217*
XGB	*1.96919*	*1.8139*	*2.09313*	*1.90769*	*2.18127*	*1.95903*

The Plot 21 visualizes actual and predicted speed with XGBoost at 15-minute time period. The distribution indicates that most of the predicted values lie near the speed

limit of the road. Even when the prediction period is 15 minutes forward the data points are in a compact cluster apart from a few exceptions.



Plot 21. TMS point 126 XGBoost 15-minute actual and predicted speed

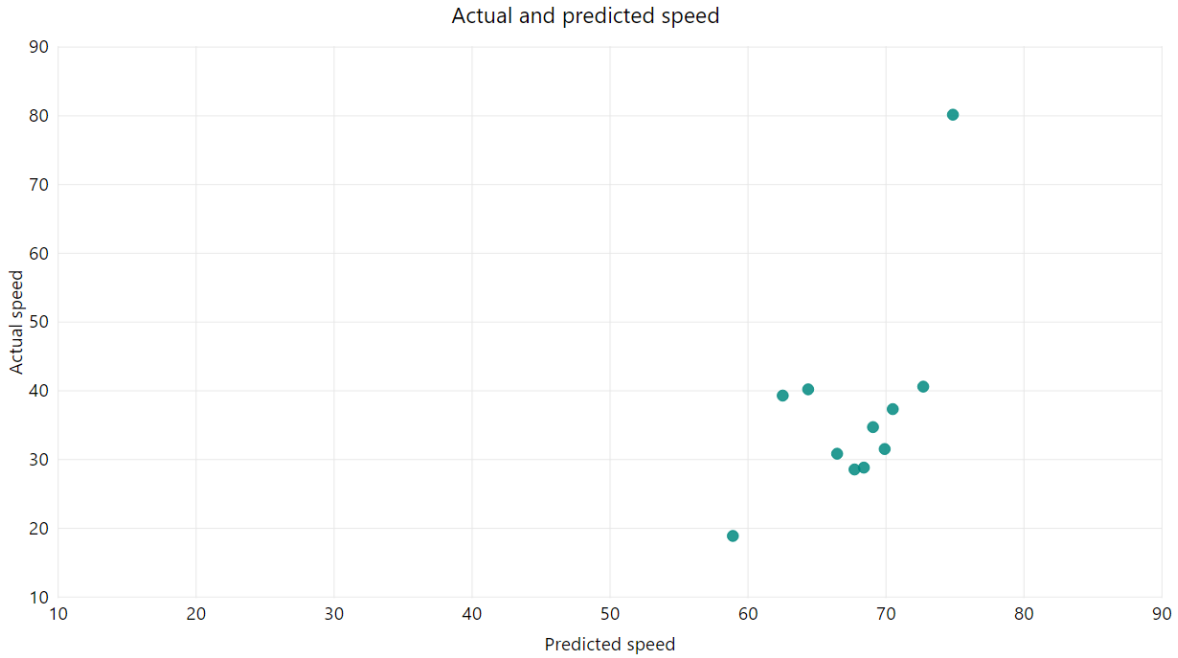
6.3.1 Speed drop comparison

The last observed TMS point consolidates the results achieved from the other two points. Time series models work best during the low-speed drops whereas when the drop increases, model predictions do not work. Extreme Gradient Boosting achieves the smallest RMSE and MAPE values when the drop is 3-5 km/h or 7-10 km/h which is comparable to the other two points. A clear abnormality comparing other results is that when the drop reaches 20 km/h or more, the best performing models are time series models.

Table 21. TMS point 126 model performance with different speed drop levels

	< 2 km/h		3-5 km/h		7-10 km/h		12-15 km/h		>= 20 km/h	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
AR(1)	1.002	1.075	3.381	4.592	7.086	10.174	11.533	17.388	26.212	63.495
ARIMA(0,1,5)	1.113	1.153	3.102	4.007	6.484	8.947	10.055	14.915	26.317	63.656
LM	1.119	1.156	3.149	4.051	6.411	8.887	10.461	16.223	30.096	92.785
KNN	1.523	1.535	3.254	3.758	6.418	8.041	10.144	14.894	32.883	101.399
XGB	1.150	1.204	2.911	3.582	5.646	7.524	10.378	15.473	26.875	82.248

With the KNN, the mean percentage error exceeds the value 100 which means that errors are greater than actual values. Based on MAPE, XGBoost and linear regression perform almost poorly as KNN. Overall, seems that no model has the predicting ability for sharp drops in the selected dataset. The plot visualizes the predicted and actual speed of the KNN model with 20 km/h or more speed drop.



Plot 22. 126 KNN drop >= 20 km/h actual and predicted values

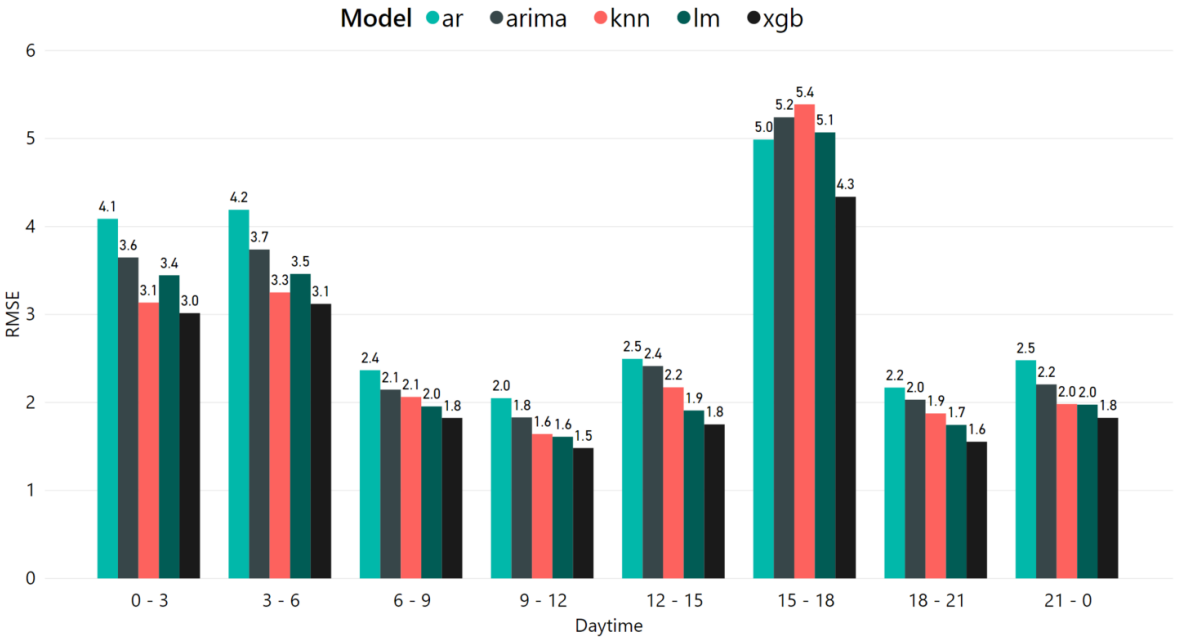
6.4 Jam classification

In the final part of the study, a decision tree is trained in order to predict the speed drop under a threshold. 40 km/h is used as a limit for “jam”, assuming that all the speed lags that are used to conduct the prediction, are above the concerned threshold. To understand the situations where the average speed reduces to less than 40km/h, it is desirable to examine the distribution of these drops during the day.

Table 22 expresses the drops in the dataset and as expected, most of the low-speed events occur during the afternoon between 15 and 17. Plot 23 offers corresponding conclusions in a case of RMSE values of the models during the day. Most of the speed variation takes place during the afternoon hours that results as more volatile prediction accuracy.

Table 22. Speed less than 40km/h at TMS point 149

Time	12	13	14	15	16	17	Total
5-minute	*1*	*2*	*14*	*141*	*62*	*8*	*228*
10-minute	*2*	*2*	*23*	*229*	*85*	*7*	*348*
15-minute	*2*	*3*	*29*	*320*	*110*	*5*	*469*



Plot 23. 5-minute prediction RMSE values at point 149 during the different daytime

Based on the distribution of the low speed, the dataset is limited to concern only data points during the afternoon hours between 15 and 17. This reduces the dataset to 7944 observations overall. Hence, the classification tree is trained for 70 percent of the filtered dataset and tested with the remaining 30 percent. 5-fold cross-validation is also performed to the decision tree model.

6.4.1 Confusion matrices and ROC

For the 5-minute prediction period, the following tree is established based on the training set (Figure 10). The root node that gives the highest contribution for the 5-minute predictions is Speed lag 1. If the lagged speed exceeds 64 km/h, the model automatically classifies the observation to the category “No jam”. Following nodes down to the bottom specifies the logic whether the observation belongs either “Jam” or “No jam”.

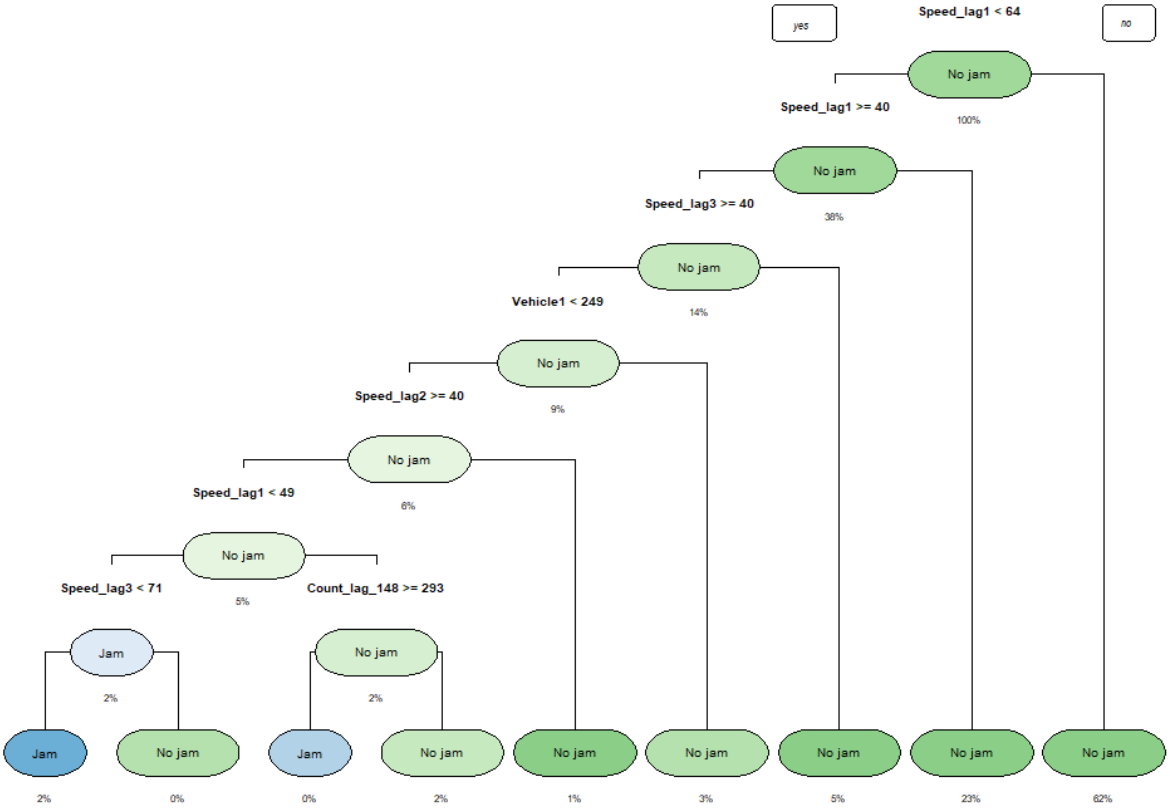


Figure 10. 5-minute prediction decision tree for jam

The established tree provides the following confusion matrix results in Table 23. The total amount of less than 40km/h speeds is 61 observations and the model predicted 32 of those correctly. Specificity of the model is, therefore, 52.45 percent as Table 24 expresses. The total amount of false negatives is 26 and true positives 2296 that indicates the sensitivity level of 98.88 %. The overall accuracy of the model is 97.69 % mainly because the number of the drops in the test set is relatively small compared to the total amount of observations.

Table 23. Confusion matrix for 5-minute predictions

	Predicted Jam	Predicted No jam	
Actual Jam	*TN = **32***	*FP = **29***	**61**
Actual No jam	*FN = **26***	*TP = **2296***	**2322**
	58	**2325**	

Table 24. Performance during different periods

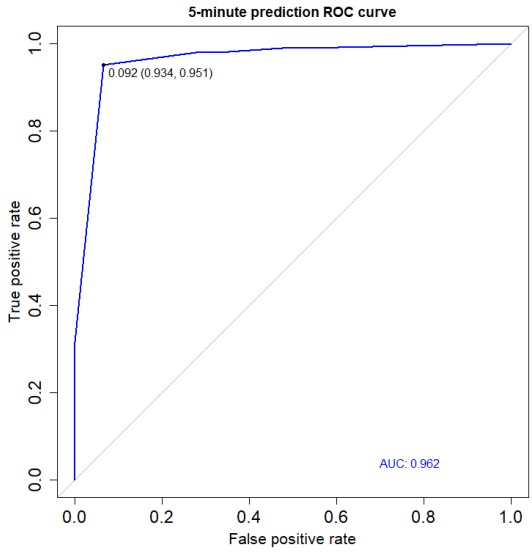
Time	Accuracy	Sensitivity	Specificity
5-minute	*0.97692*	*0.9888*	*0.52459*
10-minute	*0.96349*	*0.99126*	*0.28723*
15-minute	*0.9509*	*0.99556*	*0.17054*

To accomplish a broader understanding of the relationship between sensitivity and specificity, the ROC curve is plotted in Plot 24. The general rule is that closer the curve swipes over the left corner the better the model is. Table 25 shows the trade-off between the sensitivity and specificity using different thresholds. Thresholds are decision levels or cut-off values that generate the true positive and false positive rates when we estimate the score to belong to a class (Wang, Song & Gao, 2019) (Gigliarano et al., 2014). In this case, those refer to the probability that the score is a “Jam”. For example, using a threshold 0.092 it is possible to achieve a 95 % sensitivity level and 93 % specificity level that might indicate a successful model

depending on the requirements. Using the threshold that maximizes both values, the false positive value is 4 and the false negative is 113. Therefore, to gain the optimal level requires accepting 113 false signals of the upcoming jam and four misclassified no jam-situations, even though, there was a jam. The 100 % accuracy in jam prediction is able to achieve with a 31.5 % sensitivity level. Although, the true benefit of the classification model depends on the acceptance rates of the false positive and false negative values.

Table 25. 5-minute sensitivity-specificity trade-off with different thresholds

	Sensitivity	Specificity	Threshold
1	0	1	0
2	0.315246	1	0.001734
3	0.951335	0.934426	0.092486
4	0.979759	0.721312	0.125
5	0.979759	0.672131	0.207207
6	0.990095	0.47541	0.647059
7	0.990095	0.47541	0.664063
8	1	0	1



Plot 24. 5-minute prediction ROC curve

The increase in uncertainty is more distinct when the 10-minute and 15-minute time periods are used. Reached specificity level is less than 30 percent for both forecasting periods (Table 24) which will also result as an outstanding sensitivity

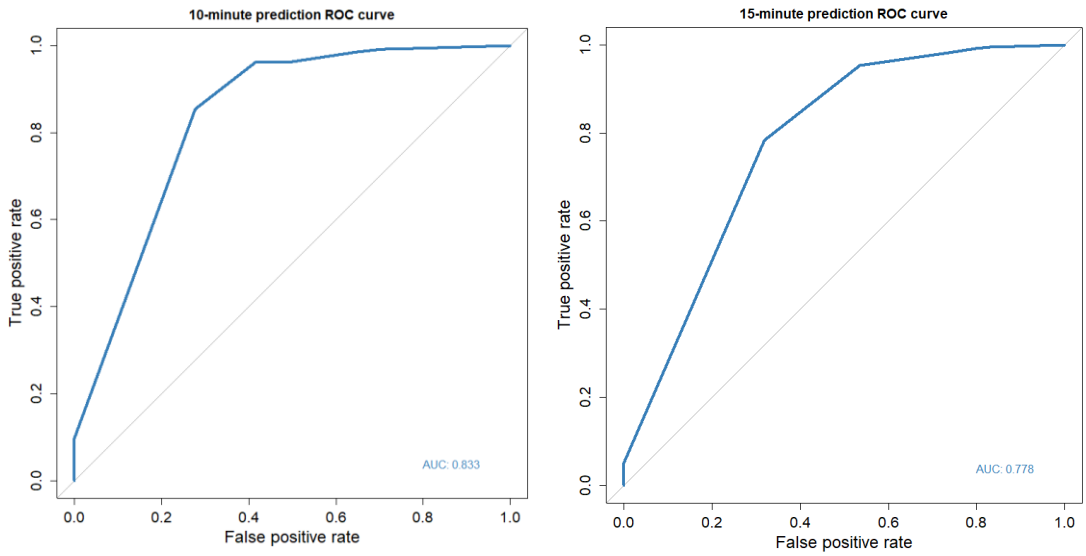
level. The misclassified jams will also reduce the overall accuracy of the classification model. Overall, the classification results during the further periods result as an imprecise prediction as the speed predicting in the first empirical part of the thesis.

Table 26. Confusion matrices for 10-minute and 15-minute predictions

	Predicted Jam	***Predicted No jam***	
Actual Jam	*TN = **27***	*FP = **67***	**94**
Actual No jam	*FN = **20***	*TP = **2269***	**2289**
	47	**2336**	

	Predicted Jam	***Predicted No jam***	
Actual Jam	*TN = **22***	*FP = **107***	**129**
Actual No jam	*FN = **10***	*TP = **2244***	**2254**
	32	**2351**	

The ROC curves for the 10- and 15-minute prediction periods are plotted in Plot 25. 5-minute prediction period provides the best threshold to gain highest sensitivity-specificity rate better results as expected, but overall the further predictions perform also at a moderate level. Thus, there seems to not have a great difference between 10 minutes and 15 minutes.



Plot 25. 10-minute and 15-minute prediction ROC curves

Overall, the problem with the classification is that the number of true negatives in the whole dataset is low. Even though it is possible to obtain relatively good performance values in a sense of specificity and sensitivity, the best threshold level in the 5-minute prediction generates 113 false signals that are almost a double times higher than the correct true negatives (61). Like stated before, the utilization of the prediction model requires selecting a certain level of accepted false negatives and if the generated accuracy is sufficient, the model is actually advisable.

7. DISCUSSION

People have tried to predict different traffic attributes for many decades, so as a topic it is not new. Although, the model approach and situations have changed during the years. Literature approves many different approaches like stated in the former studies section and as a result, it is extremely difficult to find any consensus about the “best” model for the speed predicting problems. Based on the literature and findings in the thesis, it is reasonable to state the possibility to predict short term traffic speed successfully, and in principle, every model that was studied performs at least decently in every measurement point.

Based on the study results it is clear that the XGBoost model gives the most prone results compared to other prediction models, even though the traffic conditions vary from each other. On the other hand, around 0.5-1 RMSE and MAPE difference does not seem to be substantially outstanding compared with the speed levels in question. At least, when considering the amount of data that was used in the time series models that only utilize autoregression and moving average abilities. Also, considering the training time, complexity and performance of the models, the Extreme Gradient Boosting method is clearly the most complicated one, mainly because of the number of parameters to select and tune. Variety of tuning possibilities could also lead to overfitting. Another deficiency is that the model is not interpretable. Basically, it is extremely difficult or even impossible to state how the results are generated which is not a problem with other models.

What comes to the speed drops it seems that the models are not able to capture the speed variation if it is abrupt. All models perform well in a situation were the predicted speed stays relatively the same as the lagged speed, but soon as the speed drop increases, there is no predicting ability on average. TMS 126 gave promising results providing RMSE and MAPE values close to 8 during the massive speed drops, but the significant improvement in predicting accuracy comparing the other two points comes from the wrong reasons. Although, TMS 126 high-speed drops denote an interesting comparison of how the models handle the situations where the speed increase is followed by a speed drop. At least when comparing

Linear regression to the KNN and XGBoost that usually provided similar results in the study. What comes to the traffic control systems, Wang, Yang, Liang and Liu (2018) stated that control systems for the traffic management that use induction loop technique (data in the thesis) suffer from a limited amount of data and dimensionalities related to it. Instead, more developed traffic control systems have more flexible controlling strategies and the systems are more adaptive. The future of the traffic condition predicting relies more on real-time related systems where the reaction time reduces from minutes to seconds. Thus, the expectations are that the systems utilize real-time monitoring of the data instead of forecasting traffic attributes (Gettman, Shelby, Head, Bullock & Syoke, 2007) and the systems automatically adjust the strategy of the control system rather than manual intervention (Csikos, Tettamanti & Varga, 2015).

Based on the findings, gaining more precise results in a case of speed drops might need a real-time approach, for example, GPS related systems or video image recognition. On the other hand, it is reasonable to assume that sharp speed drops occur relatively randomly and there are no clear signs to capture those in advance. For example, it is reasonable to presume that in the case of traffic accidents it is impossible to see it from the past data.

Based on the jam predicting section, it seems that even a simple decision tree can perform well during the 5-minute forecasting period. Increasing the time period to 10 or 15 minutes the uncertainty increases which leads to more unprecise predicting accuracy. In addition, based on the ROC curve, 10-minute and 15-minute predictions also provide sufficient results. Even though it is possible to obtain good results when predicting speed under 40 km/h, a success of the model relies on a question: how to define a “jam”? Interpretation of the question depends on the situation. In some cases, it might refer to the amount of traffic which means a high number of vehicles during the selected time period. For example, the density of cars will cause congestion to occur. When the density exceeds a critical limit, the speed fluctuation of the vehicles will result as an increasing fluctuation in speed of the vehicles behind it (Sugiyama, Fukui, Kikuchi, Hasebe, Nakayama, Nishinari, Tadaki & Yukawa, 2008). According to Quek and Chew (2014), traffic jams are caused by

bottlenecks which occur because of the bends on traffic highways. Although, a “jam” can also refer to the speed of the vehicles which is in the case of the study a definition for the traffic jam. The question arises when the average speed is low but also the number of vehicles is low.

Nevertheless, the success of the decision tree model during the 5-minute prediction period is that the average speed is already significantly lower than the average speed in the area. This also states the problematic of predicting sharp speed drops, because for the model there has to be some drops in the speed already.

8. CONCLUSIONS

The purpose of this study is to examine different statistical and machine learning models' ability to predict short-term traffic speed. The research covers the introduction to the literature of the traffic predicting and presents the characteristics of different predicting models that are used to conduct the speed forecasting. An autoregressive model, ARIMA model, linear regression, K-nearest neighbor and Extreme Gradient Boosted Tree were used to predict short-term traffic speed for 5, 10 and 15 minutes forward. Models' predicting abilities were also tested during the different speed drop levels. The selected speed drop levels are < 2 km/h, between 3-5 km/h, 7-10 km/h, 12-15 km/h and when the drop is >= 20 km/h. The comparison of the models is done by using Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) performance measures. Finally, the decision tree classification algorithm was trained to examine the possibility to predict traffic jams which means in this context, speed less than 40 kilometers per hour.

The traffic data is gathered from The Finnish Transportation Agency's and the weather data from The Finnish Meteorological Institute's open source platforms. There are totally three different TMS points that are used to validate the models and the selected period where the data is obtained is from the beginning of November 2017 to 27th of September 2018. Before the models can be evaluated, the data was combined and aggregated to the 5-minute averages.

The first research question is:

1. ***Which model accomplishes the best predicting accuracy of the average 5-minute traffic speed?***
    a. *Prediction period 5 minutes forward*
    b. *Prediction period 10 minutes forward*
    c. *Prediction period 15 minutes forward*

- *XGBoost outperforms other models in every measurement point and during every prediction period*

Based on the RMSE and MAPE values the XGBoost performs the best in TMS 149, TMS 107 and TMS 126 and during every forecasting period. The most successful performance figures were achieved when the forecasting is 5 minutes forward which is expected based on the existing literature. RMSE values for the points 149, 107 and 126 are 2.55, 2.12 and 1.97 and MAPE values are 2.47, 1.75 and 1.81, respectively.

According to the results, it is rather distinct that all the models that are compared in the study can be used to predict short-term traffic speed. Even though the XGBoost provides the best results compared to the other models the difference in the performance is not notable. Even the simple AR(1) performs with a decent prediction accuracy in every measurement point (during the 5-minute prediction period the worst RMSE value and MAPE value are 3.27 and 3.28).

The second research question is:

2. ***How well do the models perform during the abrupt decline of the average 5-minute traffic speed?***

- *Models do not capture well the speed variation if the drop is high.*

All three of the examination points indicated that the simple time series models perform best during the small speed drops. When the speed drop increases, models that include more dimensionalities (Linear regression, KNN and XGBoost) are able to capture the speed variation better. Although, when the magnitude of the speed drop increases, none of the examined models perform acceptably. Overall, the XGBoost generates the best results during the abrupt speed decrease, but when the drop exceeds 20 km/h or more, there is no consensus about the best performing model. In point 149 the best RMSE (19.49) and MAPE (48.18) values are generated by XGBoost, in point 107 the KNN model provided the best results (8.09 and 8.06) and in point 126 the ARIMA model is the best one (26.21 and 63.65).

Compared to the rest of the dataset, sharp drops tend to occur more or less randomly and there is no clear indicator behind the abrupt decline in traffic speed. That is one substantial reason why the models do not perform well during the sharp speed drops and therefore are not particularly usable for drop predicting at least with the datasets that are used in the study.

The third research question is:

3. ***Is it possible to predict drops of the average 5-minute speed under 40 kilometers per hour?***

- *Predicting a binary classifier, the achieved sensitivity levels for the 5-, 10- and 15-minute predictions 0.988, 0.991 and 0.995 and specificity levels are 0.524, 0.287 and 0.170. AUC values using the same forecasting periods are 0.962, 0,833 and 0.778.*

The decision tree is able to predict speed drops under 40 kilometers per hour well when the predicting period is 5 minutes forward. When the prediction time is increased to 10 or 15 minutes, the uncertainty increases which can be seen from the specificity levels. Based on the ROC curve it is possible to accomplish satisfactory results with all time periods. Overall, the model success depends on a trade-off between sensitivity and specificity. It is possible to achieve excellent specificity levels, but it will lead to accepting a certain amount (113 in the best case) of “false signals” of the jam.

What comes to the usability of the model, it depends on the situation and what is the objective that is tried to achieve. If the desired goal is to predict speed drops under the 40 km/h there is clear evidence that the model can be used for that purpose. Although, for the trained decision tree to work, it is required that the speed is already substantially lower than the average speed in the measurement point. Thus, If the goal is to predict traffic jams in advance, it is required to decide whether the information about the average speed less than 40 km/h is interesting anymore

when the speed is already substantially lower than the speed on average and close to the 40km/h threshold.

For further studies, it would be interesting to focus on predicting traffic speed with a time period that is less than five minutes. Aggregation of the data will always lead to averaging the recorded values so it would be interesting to see how the traffic behaves when the prediction period is, for example, one minute from now. This would also lead to more real-time predictions which could result as better prediction accuracy in sharp speed drops. Another extension is to add more TMS points as explanatory factors. Only the previous point was used in the study but including subsequent measurements points can generate more information about the traffic flow in a certain part of the road.

REFERENCES

Alvarez-Ramirez, J. & Rodriges, E. 2018, “AR(p)-based detrended fluctuation analysis”, *Physica A*, vol. 502, pp. 49-57.

Arlot, S. & Celisse, A. 2010, "A survey of cross-validation procedures for model selection", *Statistics Surveys,* vol. 4, pp. 40-79.

Aslanidis, N., Christiansen, C. & Cipollini, A. 2018, "Predicting bond betas using macro-finance variables", *Finance Research Letters,*

Banks, H.T. & Joyner, M.L. 2017, "AIC under the framework of least squares estimation", *Applied Mathematics Letters,* vol. 74, pp. 33-45.

Bao, Y., Ishii, N. & Du, X. 2004, “Combining Multiple k-Nearest Neighbor Classifier Using Different Distance Functions”, pp. 634-641

Bergmeir, C., Hyndman, R.J. & Koo, B. 2018, "A note on the validity of cross-validation for evaluating autoregressive time series prediction", *Computational Statistics and Data Analysis,* vol. 120, pp. 70-83.

Box, G.E.P. & Jenkins, G.M. 1970, “Time series analysis*”,* Holden-Day, San Francisco [u.a.].

Bradley, A.P. 1997, "The use of the area under the ROC curve in the evaluation of machine learning algorithms", *Pattern Recognition,* vol. 30, no. 7, pp. 1145-1159.

Breiman L., Friedman J., Olshen R. & Stone C. 1984, “Classification and Regression Trees”*,* Wadsworth Int. Group, 1984

Brooks, C. 2014, “Introductory econometrics for finance*”,* 2nd ed., Cambridge University Press, Cambridge.

Brzezinski, D., Stefanowski, J., Susmaga, R. & Szczęch, I. 2018, "Visual-based analysis of classification measures and their properties for class imbalanced problems", *Information Sciences,* vol. 462, pp. 242-261.

Chai, T. & Draxler, R.R. 2014, "Root mean square error (RMSE) or mean absolute error (MAE) – Arguments against avoiding RMSE in the literature", *Geoscientific Model Development,* vol. 7, no. 3, pp. 1247-1250.

Chang, K., Chang, Y. & Wu, G. 2018, "Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions", *Applied Soft Computing Journal,* vol. 73, pp. 914-920.

Chen, C., Wang, Y., Li, L., Hu, J. & Zhang, Z. 2012, "The retrieval of intra-day trend and its influence on traffic prediction", *Transportation Research Part C,* vol. 22, pp. 103-118.

Chen, H., Grant-Muller, S., Mussone, L. & Montgomery, F. 2001, "A Study of Hybrid Neural Network Approaches and the Effects of Missing Data on Traffic Forecasting", *Neural Computing & Applications,* vol. 10, no. 3, pp. 277-286.

Chen, T. & Guestrin, C. 2016, "XGBoost", ACM, pp. 785.

Cheng, F., Fu, G., Zhang, X. & Qiu, J. 2019, "Multi-objective evolutionary algorithm for optimizing the partial area under the ROC curve", *Knowledge-Based Systems,* vol. 170, pp. 61-69.

Cheng, S., Lu, F., Peng, P. & Wu, S. 2018, “An adaptive ST-KNN model that consider spatial heterogeneity”, *Computer, Environment and Urban Systems*. 71. P.186-198.

Chevillon, G. & Hendry, D.F. 2005, "Non-parametric direct multi-step estimation for forecasting economic processes", *International Journal of Forecasting,* vol. 21, no. 2, pp. 201-218.

Chung, W., Abdel-Aty, M. & Lee, J. 2018, “Spatial analysis of the effective coverage of land-based weather stations for traffic crashes”, *Applied Geography*. P.17-27

Cook, J.A. 2017, "ROC curves and nonrandom data", *Pattern Recognition Letters,* vol. 85, pp. 35-41.

Csikós, A., Tettamanti, T. & Varga, I. 2015, "Nonlinear gating control for urban road traffic network using the network fundamental diagram", *Journal of Advanced Transportation,* vol. 49, no. 5, pp. 597-615.

De Myttenaere, A., Golden, B., Le Grand, B. & Rossi, F. 2016, "Mean Absolute Percentage Error for regression models", *Neurocomputing,* vol. 192, pp. 38-48.

Dougherty, M.S. & Cobbett, M.R. 1997, “Short-term intern-urban traffic forecasts using neural networks”, *International Journal of Forecasting* 13. P. 21-31

Dunne, S. & Ghosh, B. 2012, Regime-Based Short-Term Multivariate Traffic Condition Forecasting Algorithm, *Journal of Transportation Engineering*. P. 455-466

Dunne, S. & Ghosh, B. 2012, "Regime-Based Short-Term Multivariate Traffic Condition Forecasting Algorithm", *Journal of Transportation Engineering,* vol. 138, no. 4, pp. 455-466.

Ertuğrul, ÖF. & Tağluk, M.E. 2017, "A novel version of k nearest neighbor: Dependent nearest neighbor", *Applied Soft Computing,* vol. 55, pp. 480-490.

Fan, J., Wang, X., Wu, L., Zhou, H., Zhang, F., Yu, X., Lu, X. & Xiang, Y. 2018, "Comparison of Support Vector Machine and Extreme Gradient Boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: A case study in China", *Energy Conversion and Management,* vol. 164, pp. 102-111.

Finnish Meteorological Institute 2018, “Havaintojen lataus”, [www document]. [Accessed 12.11.2018]. Available https://en.ilmatieteenlaitos.fi/open-source-code

Finnish Transportation Agency 2018a, “LAM-tiedot”, [www document]. [Accessed 12.10.2018] Available https://vayla.fi/avoindata/tietoaineistot/lam-tiedot#.XKQ_I5gzaUk

Finnish Transportation Agency 2018b, “LAM-tiedot”, [www document]. [Accessed 12.10.2018] Available https://vayla.fi/avoindata/tietoaineistot/lam-tiedot#.XKQ_I5gzaUk

García, S., Luengo, J. & Herrera, F. 2016, "Tutorial on practical tips of the most influential data preprocessing algorithms in data mining", *Knowledge-Based Systems,* vol. 98, pp. 1-29.

Gettman, D., Shelby, S.G., Head, L., Bullock, D.M. & Soyke, N. 2007, "Data-Driven Algorithms for Real-Time Adaptive Tuning of Offsets in Coordinated Traffic Signal Systems", *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2035, no. 1, pp. 1-9.

Gigliarano, C., Figini, S. & Muliere, P. 2014, "Making classifier performance comparisons when ROC curves intersect", *Computational Statistics and Data Analysis,* vol. 77, pp. 300-312.

Gkioulekas, I. & Papageorgiou, L.G. 2019, "Piecewise regression analysis through information criteria using mathematical programming", *Expert Systems With Applications,* vol. 121, pp. 362-372.

Goncalves, L., Subtil, A., Oliveira, M.R. & de Zea Bermudez, P. 2014, "ROC curve estimation: an overview", *REVSTAT - Statistical Journal,* vol. 12, no. 1, pp. 1.

Gou, J., Qiu, W., Yi, Z., Shen, X., Zhan, Y. & Ou, W. 2019, "Locality constrained representation-based K-nearest neighbor classification", *Knowledge-Based Systems.*

Goves, C., North, R., Johnston, R. & Fletcher, G. 2016, Short term traffic prediction on UK motorway network using neural networks, *Transportation Research Procedia* 13. p.184-195.

Goyal, R., Chandra, P. & Singh, Y. 2014, "Suitability of KNN Regression in the Development of Interaction based Software Fault Prediction Models", *IERI Procedia,* vol. 6, pp. 15-21.

Guo, J. & Williams, B.M. 2010, “Real-Time Short-Term Traffic Speed Level Forecasting and Uncertainty Quantification Using Layered Kalman Filters”, *Transportation Research Record Journal of the Transportation Research Board*. 2010. P. 28-37

Guo, S., Wu, R., Tong, Q., Zeng, G., Yang, J., Chen, L., Zhu, T., Lv, W. & Li, D. 2018, “Is city traffic damaged by torrential rain?”, *Physica A* 503. P. 1073-1080

Hyndman, R.J. & Koehler, A.B. 2006, "Another look at measures of forecast accuracy", *International Journal of Forecasting,* vol. 22, no. 4, pp. 679-688.

James, G., Witten, D., Hastie, T. & Tibshirani, R. 2013, “An introduction to statistical learning”*,* Springer, New York.

James, G., Witten, D., Hastie, T. & Tibshirani, R. 2015, “Introduction to statistical learning”*,* Corrected at 6th printing ed., Springer, New York.

Jung, B. 2013, "Exit times for multivariate autoregressive processes", *Stochastic Processes and their Applications,* vol. 123, no. 8, pp. 3052-3063.

Jung, Y. & Hu, J. 2015, "A K-fold averaging cross-validation procedure", *Journal of Nonparametric Statistics,* vol. 27, no. 2, pp. 167.

Kim, S., Kim, J. & Ryu, K.R 2016, “Comparison of Different k-NN Models for Speed prediction in an Urban Traffic Network”, *International Journal of Computer and Information Engineering*, Vol. 10, 2. p.419-422.

Kim, S., Rim, H., Oh, C., Jeong, E & Kim, Y. 2016, “Multiple-Step Traffic Speed Forecasting Strategy for Winter Freeway Operations”, *Transportation Research Record Journal of the Transportation Research Board*. P. 133-140.

Kini, B.V. & Sekhar, C.C. 2013, "Large margin mixture of AR models for time series classification", *Applied Soft Computing Journal,* vol. 13, no. 1, pp. 361-371.

Kozak, J. 2019, “Decision tree and ensemble learning based on ant colony optimization”, Springer, Cham, Switzerland.

Laña, I., Olabarrieta, I., Vélez, M. & Del Ser, J. 2018, "On the imputation of missing data for road traffic forecasting: New insights and novel techniques", *Transportation Research Part C,* vol. 90, pp. 18-33.

Lee, Y., Jung, C. & Kim, S. 2019, "Spatial distribution of soil moisture estimates using a multiple linear regression model and Korean geostationary satellite (COMS) data", *Agricultural Water Management,* vol. 213, pp. 580-593.

Ma, X., Sha, J., Wang, D., Yu, Y., Yang, Q. & Niu, X. 2018, "Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning", *Electronic Commerce Research and Applications,* vol. 31, pp. 24-39.

Maimon, O.Z. 2010, “Data mining and knowledge discovery handbook”, 2nd ed., Springer, New York.

McKenzie, J. 2011, "Mean absolute percentage error and bias in economic forecasting", *Economics Letters,* vol. 113, no. 3, pp. 259-262.

Min, W. & Wynter, L. 2010, “Real-time road traffic prediction with spatio-temporal correlations”, *Transportation Research Part C* 19 (2011). P. 606-616.

Moll, R.J., Steel, D. & Montgomery, R.A. 2016, "AIC and the challenge of complexity: A case study from ecology", *Studies in History and Philosophy of Biol & Biomed Sci,* vol. 60, pp. 35-43.

Monahan, T. 2007, “War Rooms of the Street: Surveillance Practices in Transportation Control Centers”, *The Communication Review*, vol 10, pp. 367-389

Montgomery, D., Peck, E., Vining, G. 2013, “Solutions Manual To Accompany Introduction to Linear Regression Analysis”*.* 5th ed., Wiley, US.

Natekin, A. & Knoll, A. 2013, "Gradient boosting machines, a tutorial", *Frontiers in neurorobotics,* vol. 7, pp. 21.

Park, H., Haghani, A., Samuel, S. & Knodler, M.A. 2018, “Real-time prediction and avoidance of secondary crashes under unexpected traffic congestion”, *Accident Analysis and Prevention* 112 (2018) p. 39-49

Quek, W.L. & Chew, L.Y. 2014, "Mechanism of Traffic Jams at Speed Bottlenecks", *Procedia Computer Science,* vol. 29, pp. 289-298.

Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M. & Herrera, F. 2017, "A survey on data preprocessing for data stream mining: Current status and future directions", *Neurocomputing,* vol. 239, pp. 39-57.

Rao, H., Shi, X., Rodrigue, A.K., Feng, J., Xia, Y., Elhoseny, M., Yuan, X. & Gu, L. 2019, "Feature selection based on artificial bee colony and gradient boosting decision tree", *Applied Soft Computing Journal,* vol. 74, pp. 634-642.

Ren, L. & Glasure, Y. 2009, "Applicability of the Revised Mean Absolute Percentage Errors (MAPE) Approach to Some Popular Normal and Non-normal Independent Time Series", *International Advances in Economic Research,* vol. 15, no. 4, pp. 409-420.

Ruuska, S., Hämäläinen, W., Kajava, S., Mughal, M., Matilainen, P. & Mononen, J. 2018, "Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle", *Behavioural Processes,* vol. 148, pp. 56-62.

Santra, A.K & Christy, J. 2012, "Genetic Algorithm and Confusion Matrix for Document Clustering", *International Journal of Computer Science Issues (IJCSI),* vol. 9, no. 1, pp. 322.

Sathiaraj, D., Punkasem, T., Wang, F. & Seedah, D. 2018, “Data-driven analysis on the effects of extreme weather elements on traffic volume in Atlanta”. *Computers, Environment and Urban Systems* 72. P. 212-220.

Schmidt, A.F. & Finan, C. 2018, "Linear regression and the normality assumption", *Journal of Clinical Epidemiology,* vol. 98, pp. 146-151.

Schubert, A., Hagemann, D., Voss, A. & Bergmann, K. 2017, "Evaluating the model fit of diffusion models with the root mean square error of approximation", *Journal of Mathematical Psychology,* vol. 77, pp. 29-45.

Smith, B.L., Scherer, W.T. & Conklin, J.H. 2003, *Exploring imputation techniques for missing data in transportation management systems*.

Sun, S., Huang, R. & Gao, Y. 2012, "Network-Scale Traffic Modeling and Forecasting with Graphical Lasso and Neural Networks", *Journal of Transportation Engineering,* vol. 138, no. 11, pp. 1358-1367.

Sugiyama, Y., Fukui, M., Kikuchi, M., Hasebe, K., Nakayama, A., Nishinari, K., Tadaki, S. & Yukawa, S. 2008, "Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam", *New Journal of Physics,* vol. 10, no. 3, pp. 033001.

Touzani, S., Granderson, J. & Fernandes, S. 2018, "Gradient boosting machine for modeling the energy consumption of commercial buildings", *Energy & Buildings,* vol. 158, pp. 1533-1543.

Transparency Market Research, 2019, “Global Intelligent Transportation System Market to Attain US$54.44 bn by 2024 end, Opportunities to be Fueled by Considerable Governmental Support”, [www document]. [Accessed 12.4.2019] Available https://www.transparencymarketresearch.com/pressrelease/intelligent-transportation-system-market.htm

Turner, S. M., T. Lomax, and R. Margiotta. “Monitoring Urban Roadways in 2000: Using Archived Operations Data for Reliability and Mobility Measurement”, *Texas Transportation Institute, College Station*, 2001.

Vlahogianni, E.I. 2009, "Enhancing Predictions in Signalized Arterials with Information on Short-Term Traffic Flow Dynamics", *Journal of Intelligent Transportation Systems,* vol. 13, no. 2, pp. 73-84.

Vlahogianni, E.I. 2015, "Optimization of traffic forecasting: Intelligent surrogate modeling", *Transportation Research Part C,* vol. 55, pp. 14-23.

Vlahogianni, E.I., Karlaftis, M.G. & Golias, J.C. 2014, "Short-term traffic forecasting: where we are and where we're going", *Transportation Research Part C,* vol. 43, pp. 3.

Wagenmakers, E.J. & Farrell, S. 2004, "AIC model selection using Akaike weights", *Psychonomic Bulletin & Review,* vol. 11, no. 1, pp. 192-196.

Wang, J. & Shi, Q. 2013, "Short-term traffic speed forecasting hybrid model based on Chaos–Wavelet Analysis-Support Vector Machine theory", *Transportation Research Part C,* vol. 27, pp. 219-232.

Wang, X., Song, L., Sun, L. & Gao, H. 2019, "Nonparametric estimation of the ROC curve based on the Bernstein polynomial", *Journal of Statistical Planning and Inference.*

Wang, Y., Yang, X., Liang, H. & Liu, Y. 2018, "A Review of the Self-Adaptive Traffic Signal Control System Based on Future Traffic Environment", *Journal of Advanced Transportation,* vol. 2018, pp. 1-12.

Wazakar, S., Keshavamurthy, B.N. & Hussain A 2017, “Region-based Segmentation of Social Images Using Soft KNN Algorithm”, *Procedia Computer Science 125*. P.93-98.

Weisberg, S. 2005, “Applied linear regression”, 3rd ed., Wiley, Hoboken, NJ.

XGBoost 2018, “XGBoost Parameters” [www document]. [Accessed 12.1.2019]. Available https://xgboost.readthedocs.io/en/latest/parameter.html

Xiong, D., Gui, Q., Hou, W. & Ding, M. 2018, "Gradient boosting for single image super-resolution", *Information Sciences,* vol. 454-455, pp. 328-343.

Xu, S., Shao, M., Qiao, W. & Shang, P. 2018, "Generalized AIC method based on higher-order moments and entropy of financial time series", *Physica A: Statistical Mechanics and its Applications,* vol. 505, pp. 1127-1138.

Yabe, R. 2017, "Asymptotic distribution of the conditional-sum-of-squares estimator under moderate deviation from a unit root in MA(1)", *Statistics and Probability Letters,* vol. 125, pp. 220-226.

Yu, D., Liu, C., Wu, Y., Liao, S., Anwar, T., Li, W. & Zhou, C. 2019, "Forecasting short-term traffic speed based on multiple attributes of adjacent roads", *Knowledge-Based Systems,* vol. 163, pp. 472-484.

Yu, D., Liu, C., Wu, Y., Liao, S., Anwar, T., Li, W. & Zhou, C. 2019, "Forecasting short-term traffic speed based on multiple attributes of adjacent roads", *Knowledge-Based Systems,* vol. 163, pp. 472-484.

Zhang, Y. & Haghani, A. 2014, “A gradient boosting method to improve travel time prediction.” *Transportation Research Part C* 58 (2015). P. 308-324

Zhang, Y. & Yang, Y. 2015, "Cross-validation for selecting a model selection procedure", *Journal of Econometrics,* vol. 187, no. 1, pp. 95-112.

APPENDICES

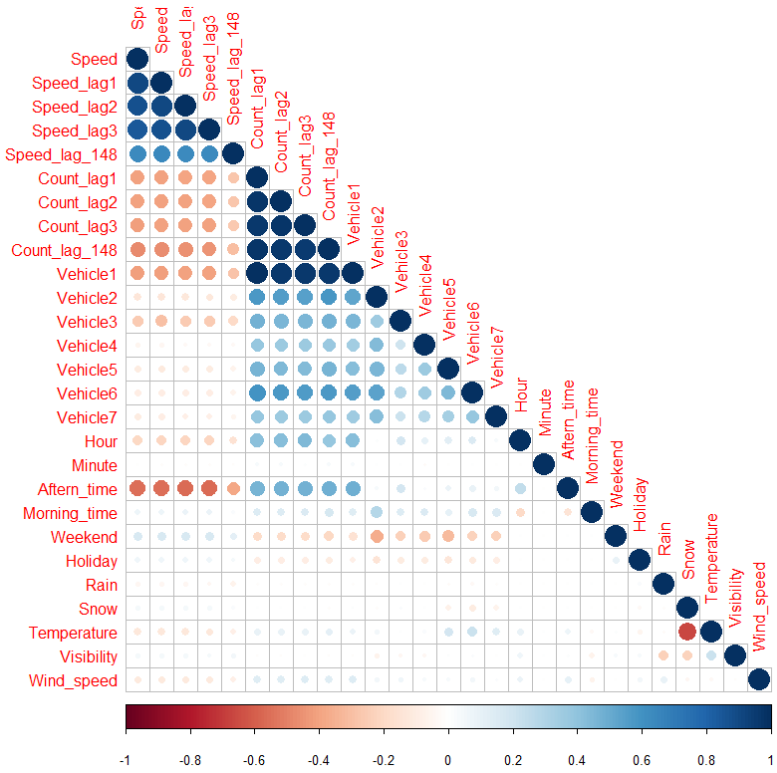
Appendix 1. TMS 107 raw data example

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16
107	18	1	0	0	35	98	3.6	1	1	1	66	0	3598	-2	0
107	18	1	0	1	29	6	3.6	1	1	1	97	0	8906	5288	0
107	18	1	0	1	31	8	3.8	1	1	1	77	0	9108	188	0
107	18	1	0	1	36	78	3.8	1	1	1	94	0	9678	552	0
107	18	1	0	1	41	87	4	1	1	1	90	0	10187	494	0
107	18	1	0	2	0	32	3.2	1	1	1	83	0	12032	1829	0
107	18	1	0	2	52	75	4.2	2	1	1	106	0	17275	-2	0
107	18	1	0	2	54	9	3.4	1	1	1	78	0	17409	5363	0
107	18	1	0	3	2	51	3.8	1	1	1	81	0	18251	826	0
107	18	1	0	3	29	96	3.8	1	1	1	82	0	20996	2728	0
107	18	1	0	4	1	8	3.6	1	1	1	80	0	24108	3095	0
107	18	1	0	4	50	60	3.8	1	1	1	96	0	29060	4935	0
107	18	1	0	4	59	71	4.2	1	1	1	81	0	29971	896	0
107	18	1	0	5	16	72	3.6	1	1	1	98	0	31672	1682	0
107	18	1	0	5	23	39	3.8	2	1	1	88	0	32339	15049	0
107	18	1	0	5	24	8	15	1	1	3	48	0	32408	722	0
107	18	1	0	6	4	13	3.6	1	1	1	78	0	36413	3892	0
107	18	1	0	6	19	14	3.8	1	1	1	85	0	37914	1484	0
107	18	1	0	7	45	94	3.8	1	1	1	81	0	46594	8663	0
107	18	1	0	7	53	26	3.6	1	1	1	99	0	47326	715	0
107	18	1	0	8	21	96	4.8	1	1	1	91	0	50196	2856	0
107	18	1	0	8	36	11	3.6	1	1	1	80	0	51611	1396	0
107	18	1	0	9	41	60	4.2	1	1	1	95	0	58160	6532	0
107	18	1	0	11	16	62	3.8	1	1	1	87	0	67662	9486	0
107	18	1	0	11	39	40	4	1	1	1	89	0	69940	2262	0
107	18	1	0	12	16	49	4	1	1	1	83	0	73649	3692	0
107	18	1	0	12	56	70	3.2	1	1	1	88	0	77670	4003	0
107	18	1	0	13	10	62	3.4	1	1	1	77	0	79062	1378	0
107	18	1	0	13	16	42	4	1	1	1	82	0	79642	564	0
107	18	1	0	14	2	70	4	1	1	1	79	0	84270	4610	0

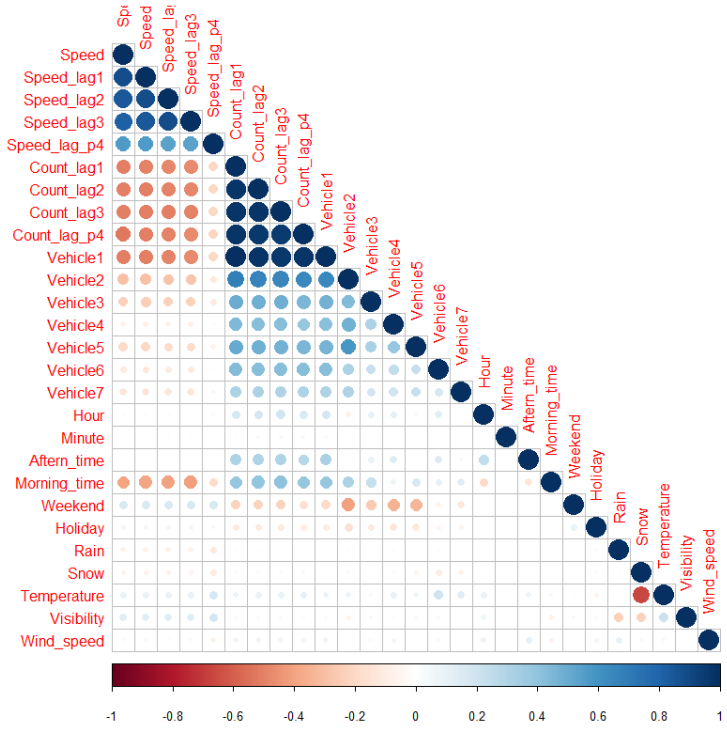
Appendix 2. Finnish national holidays during the observation period

Date	National holiday
2017-12-06	*Finnish Independence Day*
2017-12-24	*Christmas Eve*
2017-12-25	*Christmas Day*
2017-12-26	*Boxing Day*
2018-01-01	*New Year*
2018-01-06	*Epiphany*
2018-03-30	*Good Friday*
2018-04-01	*Easter Day*
2018-04-02	*2nd Easter Day*
2018-05-01	*May Day*
2018-05-10	*Ascension Day*
2018-05-13	*Mother's Day*
2018-05-20	*Pentecost*
2018-06-22	*Midsummer*
2018-06-23	*Midsummer Day*

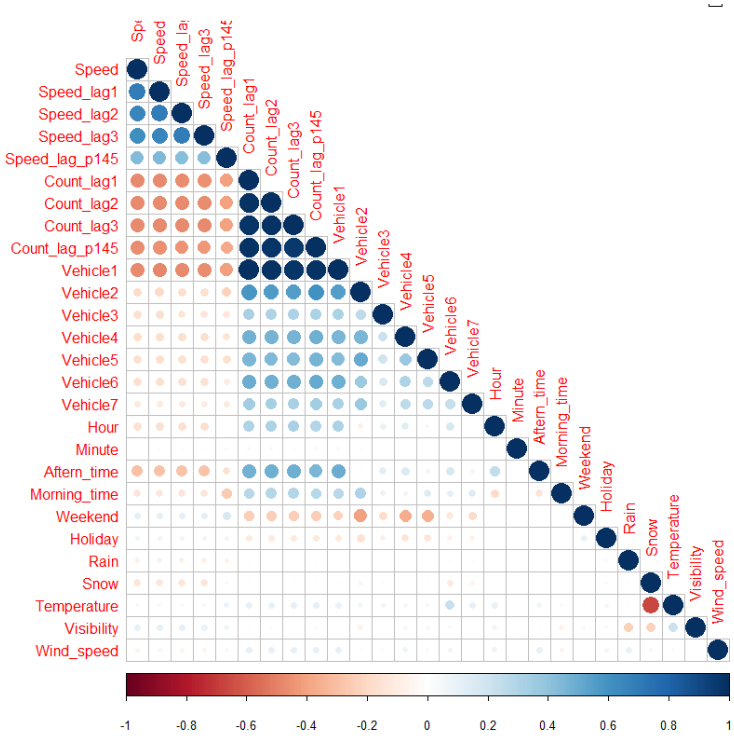
Appendix 3. Variable correlations



TMS 149 variable correlation

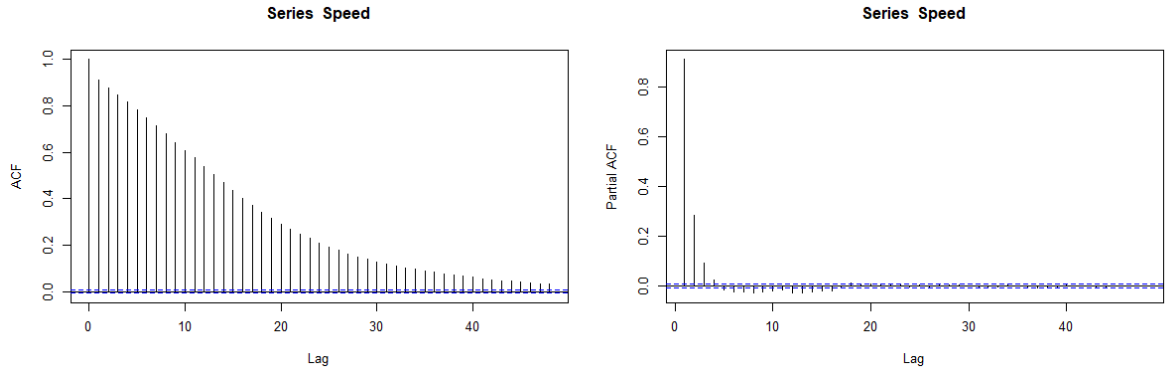


TMS 107 variable correlation

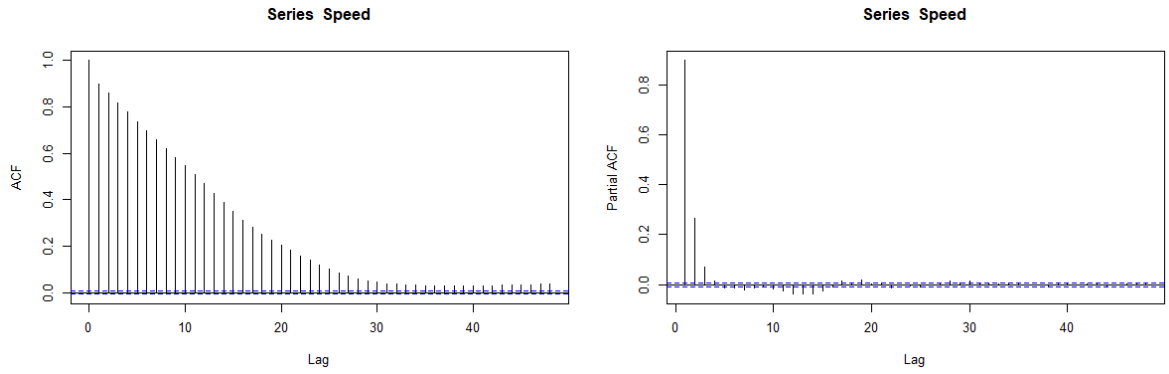


TMS 126 variable correlation

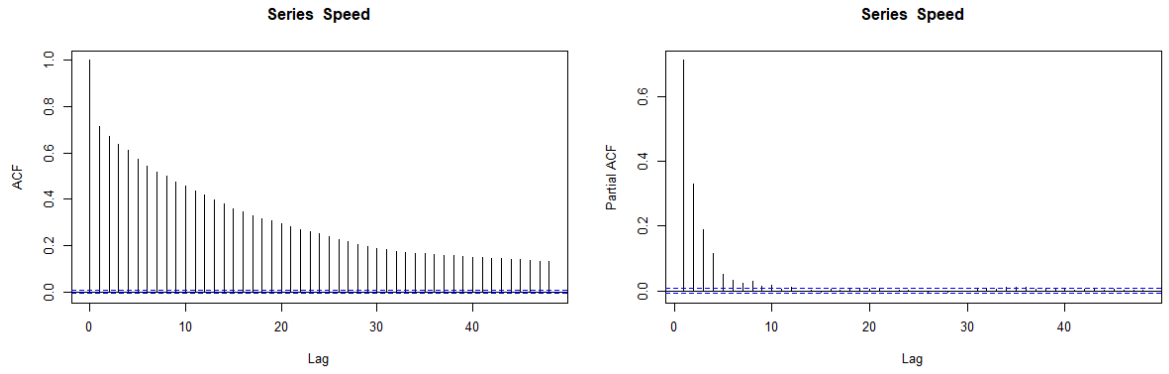
Appendix 4. Autocorrelation and partial autocorrelation functions



TMS 149 autocorrelation and partial autocorrelation



TMS 107 autocorrelation and partial autocorrelation



TMS 126 autocorrelation and partial autocorrelation

Appendix 5. Linear regression coefficients

TMS 149 10-minute linear regression coefficients

Point 149	Estimate	Std. Error	t value	Pr(>|t|)	
Intercept	29.24638152	0.33419048	87.5141073	0.00E+00	***
Speed lag 1	42.94451712	0.3774487	113.7757714	0.00E+00	***
Speed lag 2	14.50463952	0.41968228	34.5610002	1.94E-259	***
Speed lag 3	4.15974709	0.36665497	11.3451268	8.35E-30	***
Speed lag prev. point	10.73361325	0.30476507	35.2193022	3.05E-269	***
Count lag 1	64.5110731	90.35693553	0.7139582	4.75E-01	
Count lag 2	1.45959028	0.314815	4.6363429	3.55E-06	***
Count lag 3	-1.2955674	0.28331589	-4.5728724	4.82E-06	***
Count lag prev. point	-13.26646497	0.31046142	-42.731445	0.00E+00	***
Vehicle 1	-53.62829755	89.27457867	-0.6007119	5.48E-01	
Vehicle 2	-2.66371859	4.87036788	-0.5469235	5.84E-01	
Vehicle 3	-0.72444672	2.43705182	-0.2972636	7.66E-01	
Vehicle 4	-1.89121401	3.24829916	-0.5822167	5.60E-01	
Vehicle 5	-1.96012771	3.78931469	-0.5172776	6.05E-01	
Vehicle 6	-1.67743877	4.06052749	-0.4131086	6.80E-01	
Vehicle 7	-1.77854333	2.16932378	-0.8198607	4.12E-01	
Hour	0.05983152	0.05508895	1.0860893	2.77E-01	
Minute	0.27965137	0.04295426	6.510446	7.55E-11	***
Afternoon time	-1.27108007	0.05634914	-22.5572202	3.01E-112	***
Morning time	0.54432188	0.04581356	11.88124	1.60E-32	***
Weekend	0.25121303	0.03421255	7.3427161	2.12E-13	***
Holiday	0.14052097	0.06175275	2.2755418	2.29E-02	*
Rain	-2.29228988	0.98725971	-2.3218712	2.02E-02	*
Snow	0.05300903	0.06363473	0.8330203	4.05E-01	
Temperature	-0.2340519	0.09841246	-2.3782752	1.74E-02	*
Visibility	0.25044642	0.04306779	5.8151682	6.08E-09	***
Wind speed	-0.40294742	0.10008328	-4.026121	5.68E-05	***

Residual standard error: 3.45 on 66691 degrees of freedom

Multiple R-squared: 0.7939, Adjusted R-squared: 0.7939

F-statistic: 9883 on 26 and 66691 DF, p-value: < 2.2e-16

TMS 149 15-minute linear regression coefficients

Point 149	Estimate	Std. Error	t value	Pr(>|t|)	
Intercept	31.5954468	0.37248039	84.8244562	0.00E+00	***
Speed lag 1	40.3257179	0.42101472	95.7822044	0.00E+00	***
Speed lag 2	14.18039369	0.46788778	30.3072538	2.13E-200	***
Speed lag 3	0.99275924	0.40868606	2.4291488	1.51E-02	*
Speed lag prev. point	13.57833176	0.33973244	39.9677216	0.00E+00	***
Count lag 1	118.2212235	100.7424625	1.1734994	2.41E-01	
Count lag 2	1.09709916	0.35105327	3.1251643	1.78E-03	**
Count lag 3	-1.58441239	0.3152837	-5.0253545	5.04E-07	***
Count lag prev. point	-14.1390786	0.34625903	-40.8338186	0.00E+00	***
Vehicle 1	-105.6135151	99.53574985	-1.0610611	2.89E-01	
Vehicle 2	-5.67933023	5.43017038	-1.0458844	2.96E-01	
Vehicle 3	-2.36703356	2.71715465	-0.8711442	3.84E-01	
Vehicle 4	-3.79581281	3.62164774	-1.04809	2.95E-01	
Vehicle 5	-4.23148084	4.22486197	-1.0015666	3.17E-01	
Vehicle 6	-4.15578111	4.52725823	-0.9179466	3.59E-01	
Vehicle 7	-3.37177119	2.41866907	-1.3940606	1.63E-01	
Hour	0.23265025	0.06206803	3.7483103	1.78E-04	***
Minute	0.11494251	0.04778475	2.4054223	1.62E-02	*
Afternoon time	-1.86215712	0.06270608	-29.6965965	1.54E-192	***
Morning time	0.7283901	0.05127	14.2069461	9.68E-46	***
Weekend	0.3136571	0.03817055	8.2172541	2.12E-16	***
Holiday	0.19120777	0.06886727	2.7764681	5.50E-03	**
Rain	-5.19222709	1.12606021	-4.6109675	4.02E-06	***
Snow	0.03566068	0.07095471	0.5025837	6.15E-01	
Temperature	-0.37667866	0.10974764	-3.4322256	5.99E-04	***
Visibility	0.32070943	0.04805501	6.6737984	2.51E-11	***
Wind speed	-0.58326004	0.1117461	-5.2195113	1.80E-07	***

Residual standard error: 3.846 on 66690 degrees of freedom

Multiple R-squared: 0.7448, Adjusted R-squared: 0.7447

F-statistic: 7487 on 26 and 66690 DF, p-value: < 2.2e-16

TMS 107 10-minute linear regression coefficients

Point 107	Estimate	Std. Error	t value	Pr(>|t|)	
Intercept	20.6819	1.0460	19.7721	0.0000	***
Speed lag 1	57.0834	0.4914	116.1563	0.0000	***
Speed lag 2	16.3432	0.5641	28.9722	0.0000	***
Speed lag 3	3.4188	0.4860	7.0344	0.0000	***
Speed lag prev. point	16.9909	0.3491	48.6690	0.0000	***
Count lag 1	245.9788	378.5480	0.6498	0.5158	
Count lag 2	2.1228	0.4351	4.8789	0.0000	***
Count lag 3	3.5197	0.3746	9.3971	0.0000	***
Count lag prev. point	-15.7899	0.3602	-43.8319	0.0000	***
Vehicle 1	-232.9876	361.1561	-0.6451	0.5189	
Vehicle 2	-17.5442	29.6703	-0.5913	0.5543	
Vehicle 3	-5.3453	9.2087	-0.5805	0.5616	
Vehicle 4	-6.2342	12.2775	-0.5078	0.6116	
Vehicle 5	-7.2849	13.3015	-0.5477	0.5839	
Vehicle 6	-4.3273	8.1862	-0.5286	0.5971	
Vehicle 7	-2.7335	6.1421	-0.4450	0.6563	
Hour	0.4651	0.0588	7.9049	0.0000	***
Minute	0.3375	0.0476	7.0880	0.0000	***
Afternoon time	0.6842	0.0519	13.1860	0.0000	***
Morning time	-0.1990	0.0568	-3.5060	0.0005	***
Weekend	0.6892	0.0403	17.0899	0.0000	***
Holiday	0.2914	0.0713	4.0848	0.0000	***
Rain	-1.7746	1.0825	-1.6395	0.1011	
Snow	0.0788	0.0704	1.1196	0.2629	
Temperature	0.7796	0.1078	7.2293	0.0000	***
Visibility	0.1636	0.0480	3.4068	0.0007	***
Wind speed	0.1090	0.1101	0.9898	0.3223	

Residual standard error: 3.827 on 66687 degrees of freedom

Multiple R-squared: 0.7914, Adjusted R-squared: 0.7913

F-statistic: 9732 on 26 and 66691 DF, p-value: < 2.2e-16

TMS 107 15-minute linear regression coefficients

Point 107	Estimate	Std. Error	t value	Pr(>|t|)	
Intercept	24.0794	0.9862	24.4167	0.0000	***
Speed lag 1	52.1136	0.5474	95.2031	0.0000	***
Speed lag 2	15.6794	0.6286	24.9437	0.0000	***
Speed lag 3	0.2388	0.5362	0.4454	0.6560	
Speed lag prev. point	20.5845	0.3913	52.6066	0.0000	***
Count lag 1	272.5945	353.5306	0.7711	0.4407	
Count lag 2	2.7894	0.4857	5.7433	0.0000	***
Count lag 3	3.0921	0.4211	7.3421	0.0000	***
Count lag prev. point	-17.7532	0.4026	-44.0954	0.0000	***
Vehicle 1	-259.0541	337.2886	-0.7680	0.4425	
Vehicle 2	-19.2189	27.7092	-0.6936	0.4879	
Vehicle 3	-5.9476	8.6003	-0.6916	0.4892	
Vehicle 4	-6.4820	11.4661	-0.5653	0.5719	
Vehicle 5	-8.1645	12.4228	-0.6572	0.5110	
Vehicle 6	-4.6125	7.6457	-0.6033	0.5463	
Vehicle 7	-3.2520	5.7375	-0.5668	0.5709	
Hour	0.8054	0.0662	12.1636	0.0000	***
Minute	0.1747	0.0534	3.2741	0.0011	**
Afternoon time	0.9423	0.0577	16.3330	0.0000	***
Morning time	-0.1544	0.0647	-2.3867	0.0170	*
Weekend	0.8089	0.0450	17.9720	0.0000	***
Holiday	0.2707	0.0793	3.4142	0.0006	***
Rain	-4.1980	1.2625	-3.3252	0.0009	***
Snow	0.1299	0.0788	1.6480	0.0994	.
Temperature	1.0344	0.1201	8.6099	0.0000	***
Visibility	0.2303	0.0538	4.2783	0.0000	***
Wind speed	0.0743	0.1229	0.6043	0.5456	

Residual standard error: 4.271 on 66686 degrees of freedom

Multiple R-squared: 0.7402, Adjusted R-squared: 0.7401

F-statistic: 7308 on 26 and 66690 DF, p-value: < 2.2e-16

TMS 126 10-minute linear regression coefficients

Point 126	Estimate	Std. Error	t value	Pr(>|t|)	
Intercept	31.8453	0.8217	38.7566	0.0000	***
Speed lag 1	29.8055	0.4007	74.3865	0.0000	***
Speed lag 2	17.8655	0.4075	43.8450	0.0000	***
Speed lag 3	13.4598	0.3901	34.5016	0.0000	***
Speed lag prev. point	4.1502	0.1803	23.0198	0.0000	***
Count lag 1	252.2798	391.6954	0.6441	0.5195	
Count lag 2	-0.1615	0.2654	-0.6084	0.5429	
Count lag 3	1.6226	0.2288	7.0910	0.0000	***
Count lag prev. point	-4.1091	0.2540	-16.1755	0.0000	***
Vehicle 1	-242.7675	377.6226	-0.6429	0.5203	
Vehicle 2	-14.0321	22.6732	-0.6189	0.5360	
Vehicle 3	-5.3109	7.8193	-0.6792	0.4970	
Vehicle 4	-6.2470	10.1640	-0.6146	0.5388	
Vehicle 5	-6.1901	10.1639	-0.6090	0.5425	
Vehicle 6	-4.5103	7.8187	-0.5769	0.5640	
Vehicle 7	-2.7890	4.6919	-0.5944	0.5522	
Hour	0.0135	0.0342	0.3938	0.6937	
Minute	0.1572	0.0276	5.6934	0.0000	***
Afternoon time	-0.3307	0.0347	-9.5211	0.0000	***
Morning time	0.0487	0.0314	1.5500	0.1211	
Weekend	0.0455	0.0223	2.0376	0.0416	*
Holiday	0.0548	0.0396	1.3825	0.1668	
Rain	-4.7040	0.6356	-7.4010	0.0000	***
Snow	-0.5456	0.0416	-13.1128	0.0000	***
Temperature	-0.2532	0.0622	-4.0718	0.0000	***
Visibility	0.1651	0.0276	5.9818	0.0000	***
Wind speed	0.0382	0.0639	0.5976	0.5501	

Residual standard error: 2.211 on 66691 degrees of freedom

Multiple R-squared: 0.5125, Adjusted R-squared: 0.5123

F-statistic: 2697 on 26 and 66691 DF, p-value: < 2.2e-16

TMS 126 15-minute linear regression coefficients

Point 126	Estimate	Std. Error	t value	Pr(>|t|)	
Intercept	36.83329656	0.85573059	43.0430993	0.00E+00	***
Speed lag 1	28.0301008	0.41737784	67.1576167	0.00E+00	***
Speed lag 2	16.80777254	0.42436562	39.6068191	0.00E+00	***
Speed lag 3	10.53521508	0.40620474	25.9357267	1.43E-147	***
Speed lag prev. point	4.12280025	0.18788258	21.9434935	2.38E-106	***
Count lag 1	817.072822	407.9261048	2.0029922	4.52E-02	*
Count lag 2	0.30330632	0.27638834	1.0973919	2.72E-01	
Count lag 3	1.05531897	0.23843044	4.4261084	9.61E-06	***
Count lag prev. point	-4.09815801	0.26448172	-15.4950517	4.66E-54	***
Vehicle 1	-787.4679541	393.2700714	-2.0023592	4.53E-02	*
Vehicle 2	-46.82256195	23.61272191	-1.9829379	4.74E-02	*
Vehicle 3	-16.93938823	8.14328967	-2.0801653	3.75E-02	*
Vehicle 4	-20.78117183	10.58517493	-1.9632337	4.96E-02	*
Vehicle 5	-20.80611997	10.58507762	-1.9656086	4.93E-02	*
Vehicle 6	-15.91216607	8.14272501	-1.9541574	5.07E-02	.
Vehicle 7	-9.6079778	4.88632391	-1.9662998	4.93E-02	*
Hour	0.06697774	0.03590651	1.865337	6.21E-02	.
Minute	0.1049468	0.02883039	3.6401457	2.73E-04	***
Afternoon time	-0.39168506	0.03603019	-10.8710254	1.67E-27	***
Morning time	0.0776615	0.03306974	2.3484158	1.89E-02	*
Weekend	0.06041363	0.02329971	2.5928926	9.52E-03	**
Holiday	0.07003373	0.04130877	1.6953722	9.00E-02	.
Rain	-5.89482073	0.67734165	-8.7028765	3.31E-18	***
Snow	-0.63486579	0.04332744	-14.6527415	1.54E-48	***
Temperature	-0.2314013	0.06477514	-3.5723785	3.54E-04	***
Visibility	0.18077643	0.02876927	6.2836638	3.33E-10	***
Wind speed	0.02928444	0.0666699	0.4392454	6.60E-01	

Residual standard error: 2.303 on 66690 degrees of freedom

Multiple R-squared: 0.4697, Adjusted R-squared: 0.4695

F-statistic: 2272 on 26 and 66690 DF, p-value: < 2.2e-16

Appendix 6. TMS 107 5-minute XGBoost grid search