Highway Tollgates Travel Time & Volume Predictions using Support Vector Regression with Scaling Methods

Master's thesis in Computer Science: Algorithms, Languages and Logic

Amanda Yan Lin
Mengcheng Zhang

Department of Applied Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Master's thesis in
Computer Science: Algorithms, languages and logic
2017

Highway Tollgates Travel Time & Volume Predictions using Support Vector Regression with Scaling Methods

Amanda Yan Lin
Mengcheng Zhang

Department of Applied Mechanics
Chalmers University of Technology
Gothenburg, Sweden 2017

Highway tollgates travel time and volume predictions
using support vector regression with scaling methods
Amanda Yan Lin
Mengcheng Zhang

Master's Thesis 2017:79
ISSN 1652-8557
Department of Applied Mechanics
Chalmers University of Technology SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover: Traffic jam on the highway between Shanghai to Jiangsu Province during Chinese National Day in 2016. Typeset in LaTeX
Gothenburg, Sweden 2017

Highway tollgates travel time and volume predictions
using support vector regression with scaling methods
Amanda Yan Lin
Mengcheng Zhang

Department of Applied Mechanics
Chalmers University of Technology

Abstract

Toll roads or controlled-access roads are widely used around the world, for instance in Asian countries. It is often expected that drivers can drive smoother and faster on the toll roads or controlled-access roads compared to on regular roads. However, long queues happen frequently on toll roads and cause lots of problems, especially at the tollgates. Accurate predictions of travel time and volume at the tollgates are necessary for traffic management authorities in order to take appropriate measures to control future traffic flow and to improve traffic safety. This thesis describes a novel investigation on the combination of Support Vector Regression (SVR) and scaling methods for highway tollgates travel time and volume predictions. The major contribution of this thesis includes 1) an approach to handling the missing data; 2) selection of important features; 3) investigation of three scaling methods and discussion of their suitability. Experiments were done as part of the Knowledge Discovery and Data Mining (KDD) Cup 2017.

Keywords: Traffic flow prediction, traffic volume prediction, highway tollgates, time series analysis, SVR with scaling, robust scaling, SVR.

Acknowledgements

This thesis is done by Amanda Yan Lin and Mengcheng Zhang from Chalmers University of Technology in Gothenburg. We would like to thank our supervisor and examiner Dr. Selpi for her guidance in our project. She is a kind, patient and energetic person who gave us the opportunity to perform this master thesis. We are thankful for her continuous feedback on our report writing. We really enjoy the experience to work with her.
We are also grateful to SAFER for the comfortable environment for us to work on and present our thesis.
At last but not least, we would like to show our gratitude to our families and friends for their support.

Amanda Yan Lin & Mengcheng Zhang, Gothenburg, September 2017

Contents

List of Figures

List of Tables

1
Introduction

1.1 Motivation

With urbanization and motorization nowadays, problems of transportation are becoming more and more serious. Traffic jams have become common scenes in most roads, including on the toll roads. In addition, highway tollgates are well known as bottleneck in traffic networks, particularly during rush hours and special holidays. Reliable methods to predict future traffic flow are important for traffic management authorities as well as the road users. With precise predictions, the traffic regulators can decide how to deal with traffic jam or some other problems of highway tollgates (e.g.,to deploy additional toll collectors and/or divert traffic at upstream intersections). Such accurate predictions can also help road users to plan their journey.

1.2 Goals and Challenges

In this project, we address two prediction tasks, travel time prediction and traffic volume prediction, as part of a competition in Knowledge Discovery and Data Mining (KDD) Cup 2017 [1]. Travel time is a measurement of time from a designated start point to a designated end point, which is the raw element for a number of performance measures in different transportation analyzes [21]. Traffic volume are the records of the number of vehicles at a designated point. Both travel-time and volume calculations depend on lots of stochastic factors, such as weather conditions, holidays, time of the day, season, etc.
The tasks are to predict travel time and volume for a given road and tollgate during rush hours, knowing the previous two-hour data and some days before. The goal is to find suitable methods for the two predictions and to achieve good prediction performances.A big challenge is to find if those stochastic factors have effect on the predictions and how to extract appropriate features and model them in a suitable way.

1.3 Scope

The project includes exploring the use of existing algorithms to achieve the goals described in the previous section. When the performances of the algorithm are considered not so good, some adjustment may be made in data pre-processing and the algorithm. However, development of a brand new algorithm is not the purpose

of the project. The data used in both of the two tasks are from KDD Cup 2017 as well as the prediction error measurement formulas.

1.4 Thesis Outline

The rest of this paper is arranged as follows. Chapter 2 describes the two prediction tasks. Chapter 3 introduces the raw data and the data visualization. We show some theoretical backgrounds and related works in Chapter 4. In Chapter 5, We explain the methods we used. We describe and discuss the results of our experiments for travel time prediction and traffic volume prediction in Chapter 6. The conclusions are presented in Chapter 7.

2

Task description

The objectives of this thesis project are to address two prediction tasks, traffic travel-time prediction and traffic volume prediction. The road network (Figure 2.1) considered in this project includes three intersections (A, B, C) and three tollgates (1, 2, 3). Vehicles enter Intersection A can exit at tollgates 2 and 3 , while vehicles enter Intersections B and C can exit at tollgates 1 and 3. Tollgate 2 only allows traffic entering the highway, while tollgates 1 and 3 allow traffic both ways (entry and exit).
More specific, the goal is, given road network topology of the area (Fig. 2.1), vehicle trajectories, historical traffic volume at tollgates, and weather data for the area, predict travel-time and volume for the period of 25th October to 31st October. The description of given data sets are presented in Chapter 3.

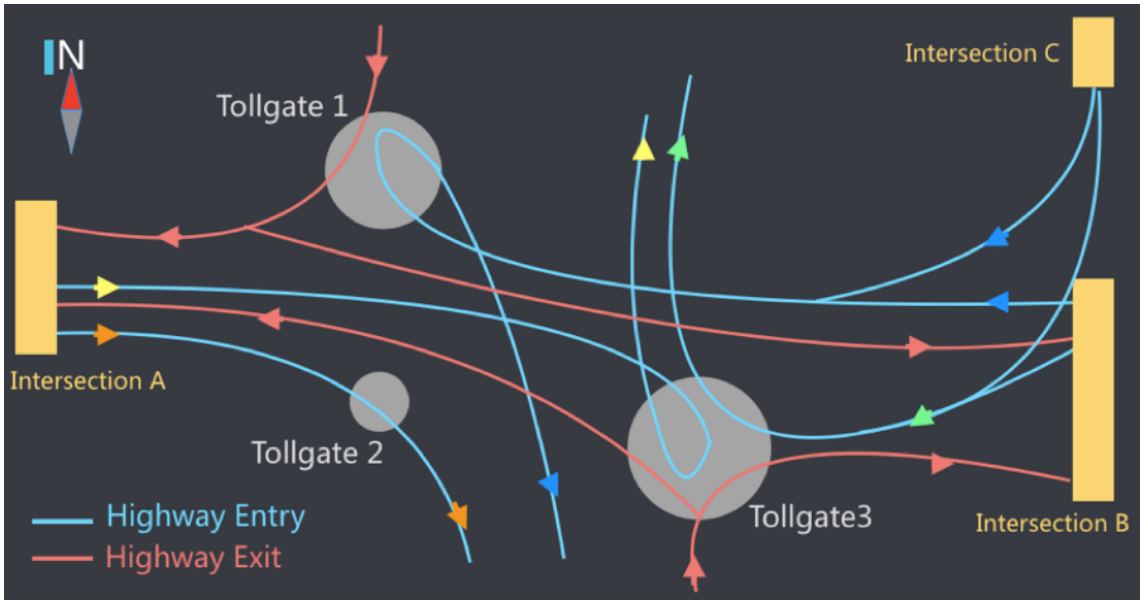

Figure 2.1: An overview of the road network. The road network consist of three intersections (A, B, C) and three tollgates (1, 2, 3). This figure is taken from the description of KDD CUP 2017 [1].

2.1 Travel-time prediction

In the travel-time prediction, the aim is to estimate the average travel time of vehicles for each route during rush hours (08:00-10:00 and 17:00-19:00), per 20 minutes

interval, for the period of 25th October to 31st October. There are totally six routes in this prediction (Figure 2.1): routes from Intersection A to Tollgates 2 and 3, routes from Intersection B to Tollgates 1 and 3, routes from Intersection C to Tollgates 1 and 3.
The Estimated Time of Arrival (ETA) of a 20-minute time window for a given route is the average travel time of all vehicle trajectories that enter the route in that time window [1]. Each 20-minute time window is defined as a right half-open interval, e.g., [2016-09-18 23:40:00, 2016-09-19 00:00:00).

2.2 Volume prediction

In the traffic volume prediction, the aim is to predict the volume for each of the five tollgate-direction pairs (Tollgate 1-entry, Tollgate 1-exit, Tollgate 2-entry, Tollgate 3-entry, and Tollgate 4-exit) during rush hours (08:00-10:00 and 17:00-19:00), per 20 minutes interval, for the period of 25th October to 31st October.
The estimated volume of a 20-minute time window for a given tollgate-direction pair is the volume of all vehicle that enter the tollgate-direction in that time window. Each 20-minute time window is defined as a right half-open interval, e.g., [2016-09-18 23:40:00, 2016-09-19 00:00:00).

3
Data

The original data was provided by organisers of the Knowledge Discovery and Data Mining (KDD) Cup 2017 [1]. Four different types of the original data set were provided: road network topology of the area (Fig. 2.1), vehicle trajectories, traffic volume at tollgates, and weather data for the area. Each of them is explained in detail in the following sections.

3.1 Road network topology

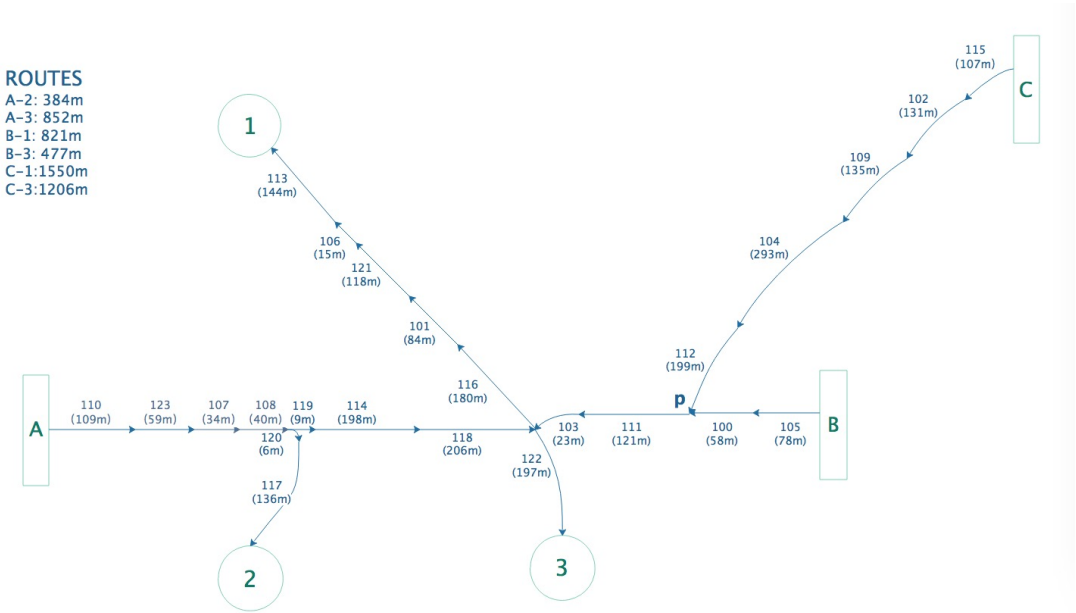

Figure 3.1: The link-representation of road network. Each route is composed by a sequence of links, each link is represented by an arrow. The value without parentheses over a link represents the unique id of the link and the value in parentheses represents the length of the link. The total length of each route is presented at the upper left corner.

The road network (Figure 2.1) used is a directed graph formed by interconnected road links, see Figure 3.1. A route in the network is composed by a sequence of links. For instance, route A-2 (Intersection A to Tollgate 2) is composed by road links: 110, 123, 107, 108, 120, 117, see Figure 3.1. For every road link, its vehicle

traffic comes from one or more ”incoming road links” and goes into one or more ”outgoing road links”.

3.2 Vehicle trajectories

Vehicle trajectories data (Table 3.1) lists time-stamped records of actual vehicles driving from intersections to tollgates. Specifically the data about vehicle trajectories consists of intersection ID, tollgate ID, vehicle ID, date time when the vehicle enters the route, trajectory (sequence of link traces with each trace consists of a link ID, time entering the link, and total travel time (in seconds) passing the link), and total travel time (in seconds) from the intersection to the tollgate.

Table 3.1: Vehicle Trajectories Along Routes

Field	Type	Description
intersection_id	string	intersection ID
tollgate_id	string	tollgate ID
vehicle_id	string	vehicle ID
starting_time	datetime	time point when the vehicle enters the route
travel_seq	string	trajectory in the form of a sequence of link traces separated by ";", each trace consist of link id, enter time, and travel time in seconds, separated by "#"
travel_time	float	the total time (in seconds) that the vehicle takes to travel from the intersection to the tollgate

Vehicle trajectories data for the period of 19th July to 24th October are provided as training data, period of 25th July to 31th October are provided as test data. The training data of vehicle trajectories consist of 24 hours time-stamped records, while the test data consist of time-stamped records between 6:00-8:00 and 15:00-17:00. Only data about vehicles using Amap navigation software was included in the vehicle trajectories data [1]. For this reason, the vehicle trajectories is only a subset of all vehicles.

3.3 Traffic volume

The data about traffic volume at tollgates (Table 3.2) consists of date time when a vehicle passes the tollgate, tollgate ID, direction (0 for entry, 1 for exit), vehicle model (integer 0 to 7 to indicate the capacity of the vehicle), boolean values indicating if the vehicle uses electronic toll collection (ETC) or not, and vehicle type (0 for passenger vehicle, 1 for cargo vehicle).
Traffic volume data for the period of 19th September to 24th October are provided as training data, period of 25th July to 31th October are provided as test data. The training data of traffic volumes consist of 24 hours records, while the test data consist of records between 6:00-8:00 and 15:00-17:00.

Table 3.2: Traffic Volume through the Tollgates

Field	Type	Description
time	datetime	the time when a vehcle passes the tollgate
tollgate_id	string	ID of the tollgate
direction	string	0 for entry, 1 for exit
vehicle_model	int	this number ranges from 0 to 7, which indicates the capacity of the vehicle (bigger the higher)
has_etc	string	does the vehicle use ETC (electronic Toll Collection) device? 0: No, 1: Yes
vehicle_type	string	vehicle type: 0 (passenger vehicle) or 1 (cargo vehicle), when a vehicle exits the highway vehicle type: NULL, when a vehicle enters the highway

3.4 Weather

The weather data (Table 3.3) consists of weather related measurements collected every three hours in the target area. Specifically the data consists of date, hour, air pressure (in hundred Pa), sea level pressure (in hundred Pa), wind direction (in degrees), wind speed (in m/s), temperature (in Celsius degrees), relative humidity, and precipitation (in mm).

Table 3.3: Weather Data (every 3 hours) in the Target Area

Field	Type	Description
date	datetime	date
hour	int	hour
pressure	float	air pressure (hPa: Hundred Pa)
sea_pressure	float	sea level pressure (hPa: Hundred Pa)
wind_direction	float	wind direction (°)
wind_speed	float	wind speed (m/s)
temperature	float	temperature (°C)
rel_humidity	float	relative humidity
precipitation	float	precipitation (mm)

Traffic weather data for the period of 1th July to 24th October are provided as training data, the period of 25th October to 31th October are provided as test data.

3.5 Illustration

There are two example figures showing some specific data characteristics of travel time data and volume data respectively. Figure 3.2 is about travel time data, it shows some outliers in the dataset, for instance the data of 8th October, 24th September, 9th October and 21st September, however, those days are not special

holidays or only normal weekdays, it is hard to conclude some common characteristics from them. Figure 3.3 is about volume data, obviously, there are two parts in the graph, the bottom part is Chinese national holidays and the upper part are normal days (not special holidays) which still have some outliers. The rest of figures about other routes and tollgates can be found in appendix A.

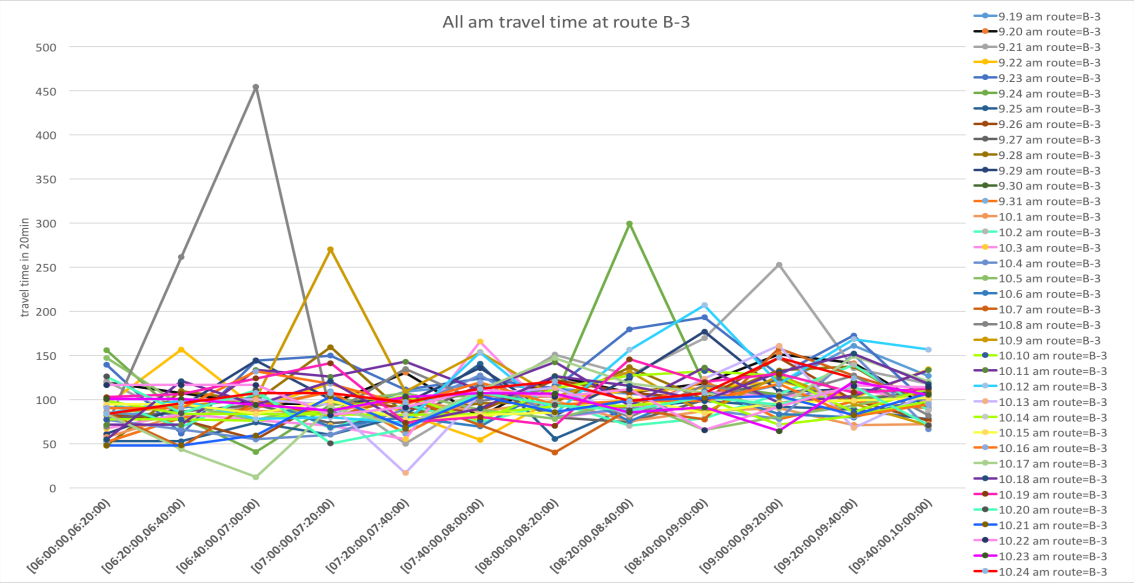

Figure 3.2: The figure shows 20-minute travel time at route B-3 (from intersection B to tollgate 3) in the morning (from 6:00 to 10:00) during the period 19th September to 24th October. The data used in the figure has been filled in by "Complementary" and linear interpolation (Section 5.1.2).

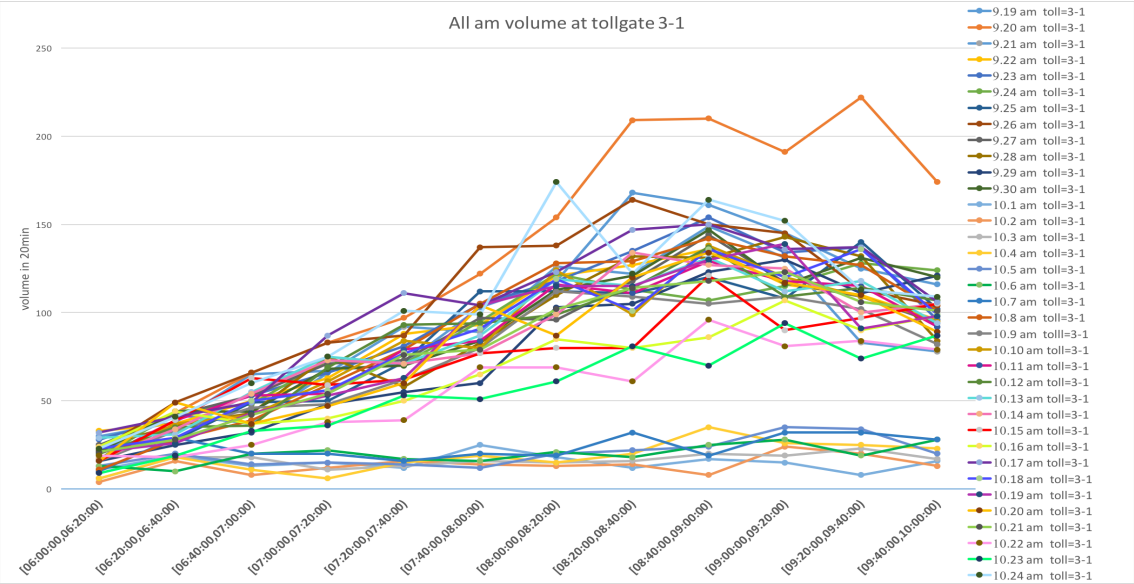

Figure 3.3: The figure shows 20-minute volume at tollgate 3-1 (tollgate 3 with direction 1) in the morning (from 6:00 to 10:00) during the period 19th September to 24th October.

4

Theoretical background and related work

In this chapter, some theoretical background about the methods used during our experiment is explained. Three scaling-methods, Min-max-scaling, standard-scaling, Robust-scaling are introduced in Section 4.1. The main algorithm used in this project, Support Vector Machine for Regression (SVR), is introduced in Section 4.2. Linear interpolation (Section 4.3) is used as a method to handle missing values and cross validation (Section 4.4) is used to assess the predictive performance of models. Finally, some previous work that shows others' attempts to solve travel-time prediction task are described in Section 4.5.

4.1 Scaling methods

Scaling is a standard step in data preprocessing and it is a way to systematically alter all the values in a data set. The simplest method, Min-Max-scaling, is rescaling the data to a fixed range, usually $[0, 1]$ or $[-1, 1]$. It is usually considered to be used for robustness to very small standard deviations of features and preserving zero entries in sparse data [3]. For a given data set X, a Min-Max-scaling is typically done via the following equation:

$$lb + \frac{X - min(X)}{max(X) - min(X)}(ub - lb),$$

where lb is a lower bound of the range, ub is an upper bound [8].
One common and widely used scaling method is Standard-scaling. If one of the features has a variance that is magnitude larger than others, it may dominate the objective function and provide bad prediction [3]. The idea of Standard-scaling is to make the values of each feature in the data have zero-mean and unit-variance (variance equal to 1), according to

$$\frac{X - mean(X)}{standard\ deviation(X)}.$$

Another scaling method is Robust-scaling, which is based on the median and the interquartile range. If the data set X contains many outliers, Robust-scaling often gives better results [4]. Robust-scaling is defined as

$$\frac{X - median(X)}{IQR},$$

where IQR is interquartile range [4].
The main advantage of scaling is to avoid features in larger numeric ranges dominating those in smaller numeric ranges. Avoiding numerical difficulties during the computation is another advantage [11].

4.2 Support Vector Machine for Regression - SVR

SVR is a version of SVM for regression that was proposed in 1996 by Vladimir N. Vapnik, Harris Drucker, Christopher J. C. Burges, Linda Kaufman and Alexander J. Smola [10]. SVR uses the same principles as the support vector machine for classification (SVC). It is an application of SVM (Support Vector Machine) for time-series forecasting [21]. SVR has shown some good performances in different areas, such as financial time series forecasting [15], stock market price forecasting [23] and real-time flood stage forecasting [24]. It was applied for travel-time prediction and achieved good result as well [21].
The goal of SVR is to find a function that best fits the data by solving an optimization problem. The model produced by SVR depends only on a subset of the whole training set. The points belongs to the subset are called Support Vectors. In order to run SVR, a kernel and several SVR-parameters have to be set (see Section 4.2.2 and Section 4.2.3).

4.2.1 SVR algorithm

Let $\{(x_1, y_1), ..., (x_l, y_l)\} \in \mathcal{X} \times \mathcal{R}$ denote a training set with training data $x_i \in \mathcal{X}$, target $y_i \in \mathcal{R}$ and size of training set l. The basic idea of SVR is to find a function $f(x)$, such that for every training data x_i, the deviation between function output $f(x_i)$ and the actual target y_i is at most ϵ. At the same time, the function $f(x)$ should be as flat as possible. We start with a simple case for linear function $f(x)$ taking the form

$$f(x) = \langle w, x_i \rangle + b$$

with $w \in \mathcal{X}$, $b \in \mathcal{R}$. The $\langle ., . \rangle$ denotes the dot product in $\mathcal{X}$. The problem can be written as a convex optimization problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \|w\|^2 \\ \text{subject to} \quad & y_i - \langle w, x_i \rangle - b \leq \epsilon \\ & \langle w, x \rangle + b - y_i \leq \epsilon \end{aligned}$$

If the problem is not feasible, slack variables ξ_i, ξ_i^* are introduced. The formulation becomes

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1} (\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i \\ & \langle w, x \rangle + b - y_i \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned}$$

where the constant $C > 0$ is a penalty parameter and determines the flatness of f. This above optimization problem can be transformed into the dual problem and its solution is given by

$$w = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) x_i \tag{4.1}$$

and

$$f(x) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b, \tag{4.2}$$

with $\alpha_i, \alpha_i^* \in [0, C]$. Note that, the complete algorithm can be described in terms of dot products between the data [5].
The algorithm can be made nonlinear by introducing a mapping function $\Phi : \mathcal{X} \rightarrow \mathcal{F}$. The idea is to map the training data from the input space $\mathcal{X}$ into a higher dimensional feature space $\mathcal{F}$ via the function Φ. Then, construct the linear model in this feature space,

$$f(x) = \langle w, \Phi(x) \rangle + b \tag{4.3}$$

As mentioned in the previous paragraph, the algorithm only depends on dot products between the data. Hence, the solution in generally can be written as

$$f(x) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) k(x_i, x) + b, \tag{4.4}$$

where

$$k(x_i, x) = \langle \Phi(x_i), \Phi(x) \rangle. \tag{4.5}$$

The function $k(x_i, x)$ is a kernel function and defined as a linear dot product of the nonlinear mapping. According to the solution (Equation 4.4 and 4.5), it is suffices to know the kernel $k(x_i, x)$ rather than Φ explicitly. The kernel computation is cheaper than explicit computation which involves computations in higher dimensional space [5].

4.2.2 SVR kernels

As mentioned previously in the end of Section 4.2.1, the kernel function $k(x_i, x)$ is used to replace the dot product $\langle \Phi(x_i), \Phi(x) \rangle$, as a result, it enables the performance of dot product without knowing the transformation Φ.
There exist several common kernel functions, for example Linear kernel, Polynomial kernel and Radial Basis Function (RBF) (Table 4.1). The RBF is commonly used as the kernel for regression [21]. After testing with several experiments, RBF was chosen as the kernel function in our studies.

4.2.3 SVR parameters

There are three parameters C, ϵ and γ that must be set when applying SVR with RBF kernel. The parameter C controls penalties of deviations between estimated values and actual target values. A low C value means low penalties, a large C value means high penalties. If C goes to infinity, SVR would not tolerate any error and

Table 4.1: Common kernel functions

Kernel	Function
Linear	$x * y$
Polynomial	$[(x * x_i) + 1]^d$
Radial Basis Function (RBF)	$exp\{-\gamma\|x - x_i\|^2\}$

create a complex model, whereas if C goes to 0, lots of errors will be tolerated and the model would be less complex [21].
The parameter ϵ is in charge of the range of the ϵ-insensitive zone (Fig 4.1). The data in a ϵ-radius tube will be disregarded in regression. The value of ϵ will influence the amount of data that can be used to construct the regression function. As a result, larger value of ϵ can make regression model less complex [7].

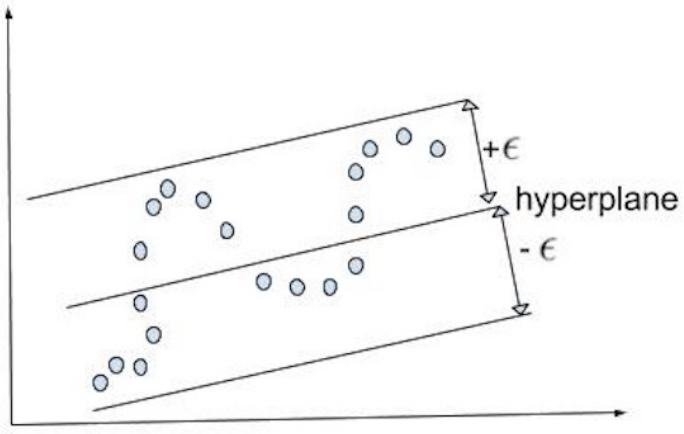

Figure 4.1: A ϵ-radius tube to the data in SVR. The figure was adapted from [21] .

The parameter γ comes from RBF kernel function

$$k(x_i, x) = exp(-\frac{|x - x_i|^2}{2p^2})$$

where parameter p is the width parameter, $\gamma = \frac{1}{2p^2}$.
The selection of parameter p depends on the input range of the training/test data set [7].

4.3 Linear interpolation

Linear interpolation is a method to calculate the approximate value of a function $f(x)$, it replaces the value of $f(x)$ by a linear function:

$$L(x) = a(x - x_1) + b,$$

at given points x_1 and x_2, $L(x)$ has the same value with $f(x)$, as a result the parameters a and b can be chosen by:

$$L(x_1) = f(x_1), L(x_2) = f(x_2).$$

A unique function can satisfied this condition:

$$L(x) = \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1) + f(x_1),$$

which can calculate the approximate value of the given function $f(x)$ on the interval $[x_1, x_2]$. The calculation can be done by hand easily, as a result linear interpolation is widely applied to tabular data [2].

4.4 Cross validation

Cross validation was used to assess the predictive performance of our models. The concept of our cross validation method is: partition the original data set into n subsets (in our case, n is number of weeks of original data set and each subset consists of data from the same week), retain a single subset (one weeks' data) as the validation data for testing the model, using the remaining $n - 1$ subsets as training data. The cross validation process is repeated n times, such that each of the n subset used exactly once as the validation data. The n results obtained from the n cross-validation processes is averaged to provide a single estimation.

4.5 Related work

Traffic flow prediction, as a well-known problem in traffic network, has been studied in previous research.
For the travel-time prediction, both statistical (data-driven) and analytical approach (model-based) had been tried [21]. The statistical approach uses time series data consisting traffic variables such as travel times, speeds, and volumes as input and predict the current travel time based on historical traffic patterns. This approach assumes that the current (or near future) travel time will have similar pattern as historical travel time. Different from the statistical approach, the analytical approach deduces the travel time from traffic conditions. The traffic conditions in turn is predicted from traffic propagation on the network by using traffic simulators (such as NETCELL [6], and MITSIM [22]). Statistical approach is suitable to be used when there are good amount of historical data while analytical approach can be applied to the situation with changes in input factors, for example, adding additional networks [18] . Compared with analytical approach, an obvious advantage of statistical approach is that there are lots of ready-to-use software packages, the approach do not need much expertise about traffic flow modeling [20].
Support Vector Machine for Regression belongs to statistical approach and is a data-driven method. An application of SVR for highway travel-time prediction has been studied by Wu et al. in [21]. In their study, they used a sequence of

historical travel-time data $(TT(t-n), ..., TT(t-1))$ to predict the travel-time $TT(t)$ at time t. Besides, they compared three different methods: SVR Prediction Method, Current Travel-Time Prediction Method and Historical Mean Prediction Method.In the Current Travel-time prediction Method, the distance of the road divided by speed at the beginning is used to compute the travel time. For Historical Mean Prediction Method, the travel time is defined by the average travel time from the previous records of the same time of day and the same day of week. The best result was achieved by using SVR Prediction Method.
There exist two main differences between data-sets in Wu et al.'s paper and in our project, one is that they collected the data from different highways while our data were collected between different intersections and tollgates, the other is in our data we have special holidays and lots of missing data, but they avoided special holidays and set the data loss rate within some threshold value. In addition, we use feature scaling as a data pre-processing step which was not included in Wu et al.'s work.

5
Methods

5.1 Data preparation

The original data provided by KDD (described in Section 3) cannot be used directly in the prediction process. This is because the provided data is time-stamped records, but the tasks ask for 20-minute prediction and that there are missing values. Here, a conversion of the data from the original data format into a format adapted to the prediction process is called data-transformation. The data-transformations for both travel-time data and volume-data are presented in Section 5.1.1. After data-transformation, we need to deal with the data-incompleteness. How we deal with data-incompleteness is described in Section 5.1.2.

5.1.1 Data transformations

Since the tasks ask for 20-minute prediction, in order to conveniently use the data during prediction process, we transform the original vehicle trajectories data into a data set consist of average travel time for every 20-minute time window (Table 5.1). Analogously, we transform the original volume data into a data set consist of volume of a 20-minute time window for each tollgate-direction pair (Table 5.2).

Table 5.1: Average travel time of 20-minute time window

Field	Type	Description
intersection_id	string	intersection ID
tollgate_id	string	tollgate ID
time_window	string	20-minute time window, e.g., [2016-09-18 23:40:00, 2016-09-19 00:00:00)
avg_travel_time	float	the average travel time (in seconds) of all vehicle trajectories that enter the intersection and exit the tollgate in that time window

5.1.2 Missing data

As mentioned in Section 3.2, only data about vehicles using Amap navigation software were included in the vehicle trajectories data [1]. If there is not a single vehicle using Amap navigation software enters a route in a 20-minute time window, the average travel time (obtained by data transformation and described in Section 5.1.1)

Table 5.2: Traffic Volume of 20-minute time window

Field	Type	Description
tollgate_id	string	ID of the tollgate
direction	string	0 for entry, 1 for exit
time_window	string	20-minute time window, e.g., [2016-09-18 23:40:00, 2016-09-19 00:00:00)
volume	int	number of vehicles that pass the tollgate in that time window
no_vehicle_model_x	int	number of vehicles with vehicle model x (x ∈ [0,7]) that pass the tollgate in the 20-minute time window
no_etc_0	int	number of vehicles not use ETC that pass the tollgate in that time window
no_etc_1	int	number of vehicles use ETC that pass the tollgate in that time window

for the route in that 20-minute time window will be NULL. In this case, we say the value is missing for the route in that 20-minute time window. The missing values exist mainly in route B-1, B-3, C-1 and C-3. Complementary-method, introduced in the following section, is a method to fill in part of the missing values in the data set. Another simple and common way of handling missing values is linear interpolation (described in Section 4.3). In this project, we present a method by combining Complementary-method with linear interpolation (described in 5.1.2.2).

5.1.2.1 Complementary

"Complementary" is a method that we develop to estimate the missing values in the average travel time data set. The basic idea of Complementary-method is: if there is a missing value for route R (R ∈ [B-1, B-3, C-1, C-3]) in a 20-minute time window, this missing value will be filled in by the relevant part of adjacent routes of route R. The adjacent routes of route C-1 are route C-3 and B-1, the adjacent routes of route C-3 are route C-1 and B-3, the adjacent routes of route B-1 are route B-3 and C-1, the adjacent routes of route B-3 are route B-1 and C-3.
If the value for route C-3 in time window: [2016-09-18 07:00:00, 2016-09-18 07:20:00) is missing, we gather part of data for that time window from route C-1 to get Intersection C to point p (C → p) and part of data from route B-3 to get point p to Tollgate 3 (p → 3) to fill the missing value in C-3 (see Fig. 3.1). Analogously, if the value for route C-1 in time window: [2016-09-18 07:00:00, 2016-09-18 07:20:00) is missing, we gather part of data for that time window from route C-3 to get Intersection C to point p (C → p) and part of data from route B-1 to get point p to Tollgate 1 (p → 1) to fill the missing value in C-1. Similar ways were done for the routes B-1 and B-3.
There exist some limitations in Complementary-method,

1. Complementary-method can only be applied for route B-1, B-3, C-1 and C-3, not A-2 and A-3.
2. The missing values of routes B-1, B-3, C-1 and C-3 in average travel time data

set cannot be totally filled in by only using Complementary-method. Following our earlier example in the previous paragraph, if the value for route C-3 in time window: [2016-09-18 07:00:00, 2016-09-18 07:20:00) is missing, we estimate the missing value with the help of data from adjacent routes C-1 and B-3 in that time window. However, if the data from any of the adjacent routes C-1 or B-3 is missing, the Complementary-method cannot be applied. Consequently, the missing values of routes B-1, B-3, C-1 and C-3 in average travel time data set cannot be totally filled in by only using Complementary-method.

5.1.2.2 Complementary combined with linear interpolation

As previously explained in Section 5.1.2.1, the missing values in 20-minute Average Travel Time data cannot be totally filled in only using Complementary-method. This motivated us to use Complementary-method combined with linear interpolation and it is described in two steps,

1. Apply Complementary-method to fill in the missing values in routes B-1, B-3, C-1 and C-3.
2. Apply linear interpolation to fill in the rest of missing values in all routes (including routes A-2 and A-3).

Following the procedure described above, the missing values in 20-minute Average Travel Time data can be completely filled.

5.2 Error measurements

Mean Absolute Percentage Error (MAPE) has been chosen by KDD cup team to evaluate the predictions made.
For Task 1 (travel-time prediction), the MAPE is defined

$$MAPE_{travel-time} = \frac{1}{R}\sum_{r=1}^{R}(\frac{1}{T}\sum_{t=1}^{T}|\frac{d_{rt}-p_{rt}}{d_{rt}}|) \tag{5.1}$$

In the Eq. 5.1 above, d_{rt} and p_{rt} are the actual and predicted average travel time for route r during time window t.
For Task 2 (volume prediction), the MAPE is defined:

$$MAPE_{volume} = \frac{1}{M}\sum_{m=1}^{M}(\frac{1}{T}\sum_{t=1}^{T}|\frac{f_{mt}-p_{mt}}{f_{mt}}|). \tag{5.2}$$

In the Eq. 5.2, M is the number of tollgate-direction pairs (1-entry, 1-exit, 2-entry, 3-entry and 3-exit), T is the number of time windows in the testing period, and f_{mt} and p_{mt} are the actual and predicted traffic volume for a specific tollgate-direction pair m during time window t.

5.3 SVR with Scaling

As mentioned before (Section 4.5), there have been previous research using SVR-predictor for highway travel-time prediction. Due to the successful prediction result

in the previous research, we choose to focus on the SVR-predictor for both travel-time and volume prediction tasks in this project. However, before we apply SVR on these tasks, we scale the data in advance. The methodology is called SVR with Scaling in our project. It is a modification of previous work mentioned in Section 4.5. In support vector machines, feature scaling can reduce the time to find support vectors and changes the Support Vector Machine result [12]. In this project, we analyze SVR with Min-max-scaling, Standard-scaling and Robust-scaling (see Section 4.1 for description of each scaling-method).

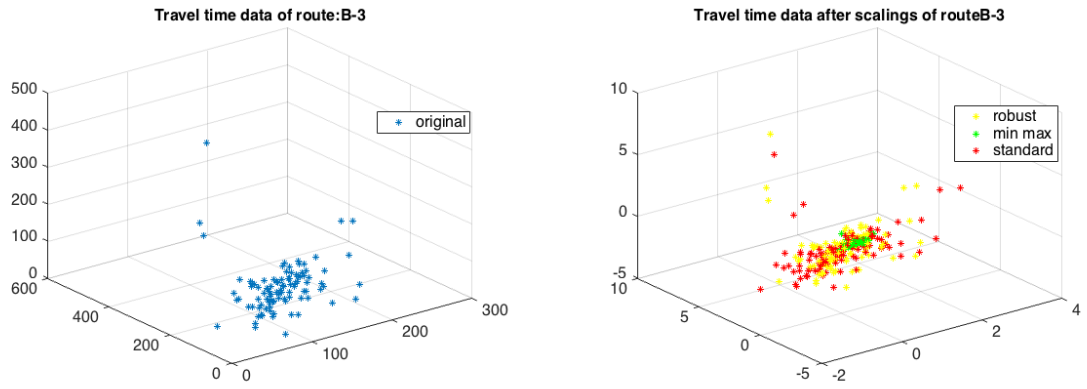

Figure 5.1: Comparison between original travel data (left figure) and travel data after scaling (right figure).

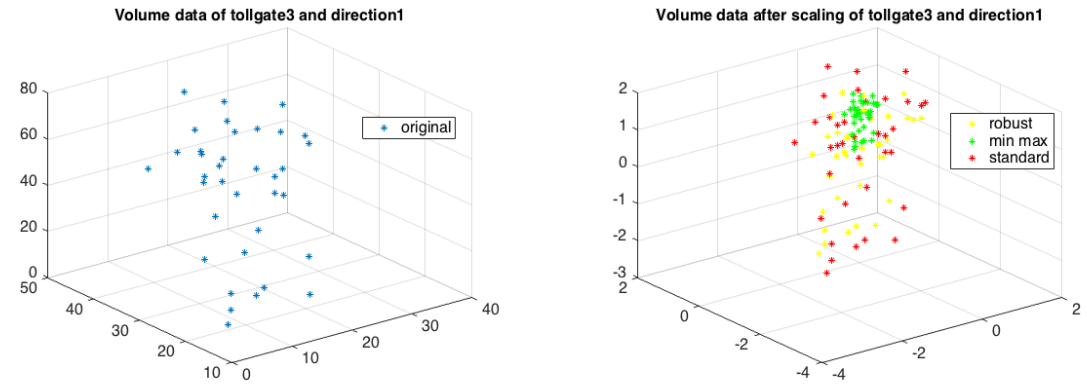

Figure 5.2: Comparison between original volume data (left figure) and volume data after scaling (right figure).

The characteristic of scaling is to transform all values into a smaller range. In particular, the Min-max-scaling method transforms all values into range [0,1]. A comparison between original travel time data and data after scaling is shown in Fig 5.1, while the comparison between original volume data and volume data after scaling is shown in Fig 5.2. Obviously, the range of data after scaling is much smaller than the range of original data.

5.4 Experimental procedure

In order to build a good model for Task1 and Task2, we address the following sub-problems:

1. Sub-problem for Task1: given training data for the period of 19th July to 17th October, estimate the average travel time, per 20 minutes interval, from designated intersections to tollgates during rush hours (08:00-10:00 and 17:00-19:00) for the period of 18th October to 24th October.
2. Sub-problem for Task2: given training data for the period of 19th September to 17th October, estimate the volume for each of the tollgate-direction pairs, per 20 minutes interval, during rush hours (08:00-10:00 and 17:00-19:00) for the period of 18th October to 24th October.

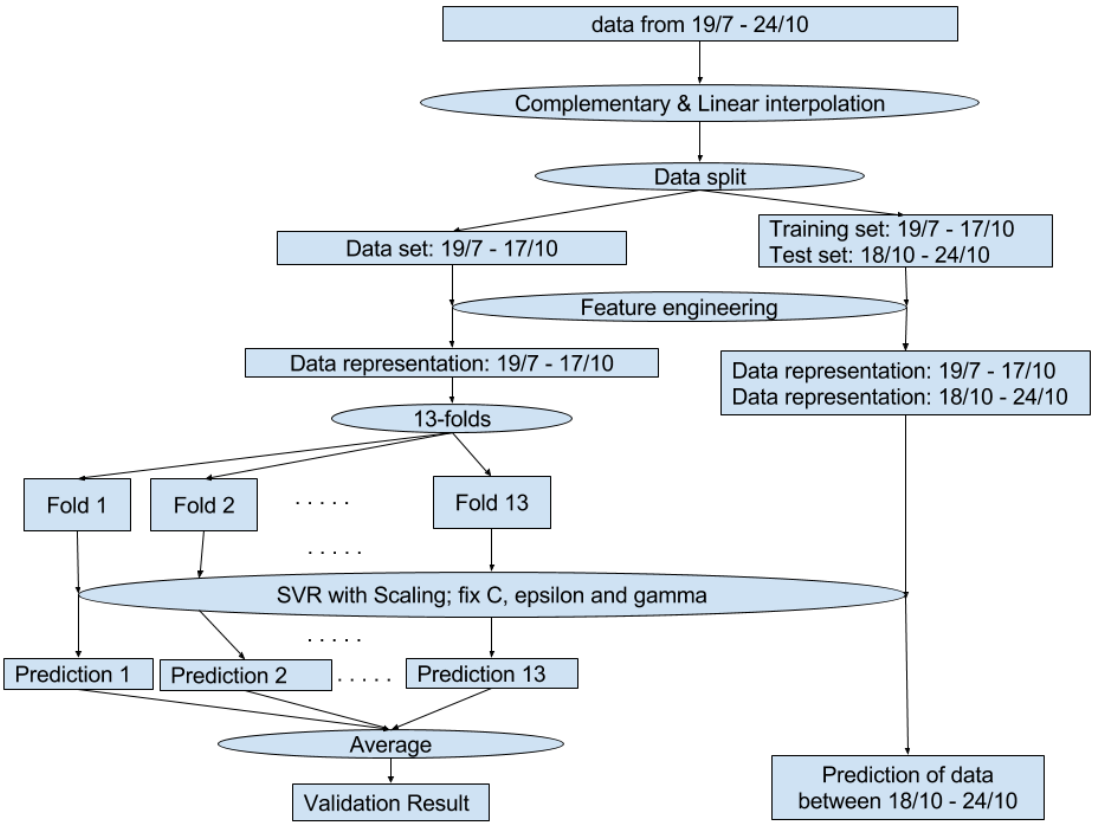

Figure 5.3: A flow chart of overall prediction model for travel time prediction.

The procedure to address those two sub-problems is quite similar and presented step by step in following text. A flow chart of overall prediction process for sub-problem 1 is presented in Figure 5.3, for sub-problem 2 is presented in Figure 5.4.

1. Handle missing values for travel time data. Using Complementary combined with linear interpolation (see Section 5.1.2.2) to fill in missing values for all routes. This step is not necessary for sub-problem 2, since there are not many missing records in volume data.
2. Split the data into two data sets, e.g training set including data for the period of 19th July to 17th October and test set including data for the period of 18th

October to 24th October.

3. Decide a feature set to investigate and form a data representation.
4. Decide a SVR-setting (γ, ϵ, C values) and a scaling-method to investigate. Perform cross-validation on data for the period of 19th July to 17th October with the particular setting. Perform a prediction on data for the period of 18th October to 24th October with same setting.

The main differences in the prediction processes between two sub-problems are: 1. we do not need to deal with the missing values for volume prediction, since there is not missing values in volume data; 2. the input data for volume prediction (volume data) is between 19/9 and 24/10, while the input data for travel time prediction (travel time data) is between 19/7 and 24/10.

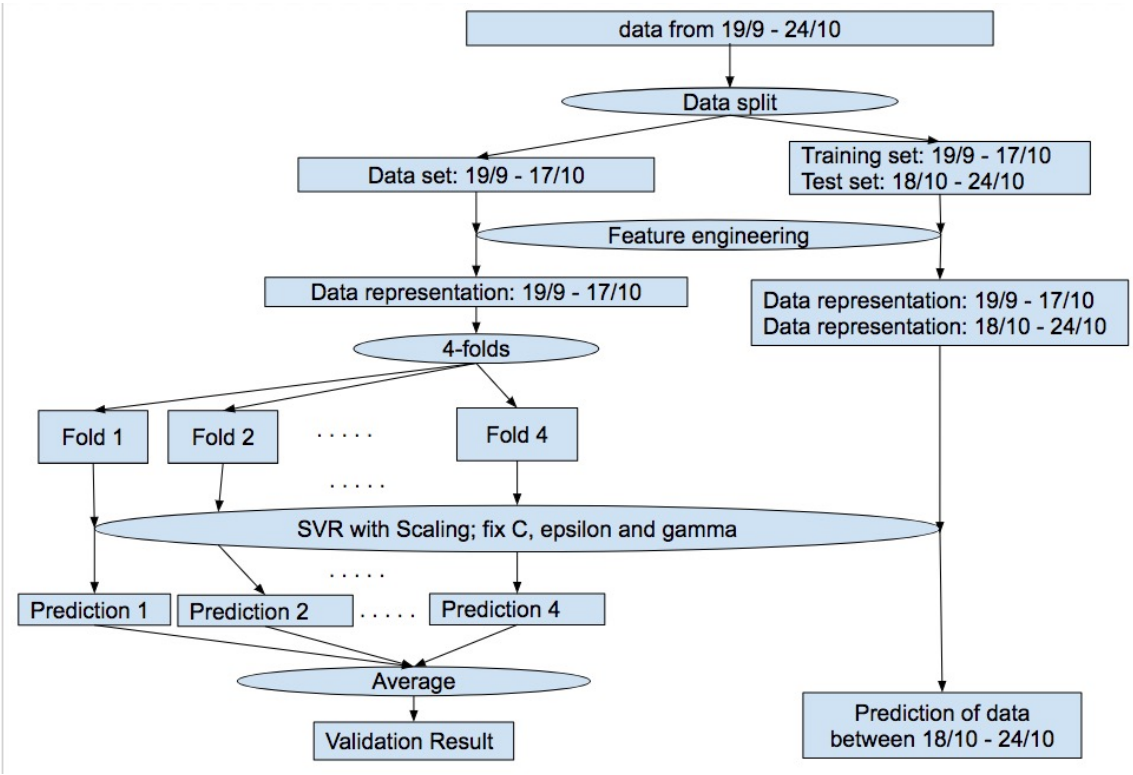

Figure 5.4: A flow chart of overall prediction model for volume prediction.

To make the model work, we have to choose some settings including feature-set, SVR paramters and scaling methods. Obviously, the validation and prediction results vary for different settings. In this project, we chose to investigate three scaling-methods: robust, standard and min-max. The results and discussion around the results is presented in the following chapter. Even the choice of feature set and SVR parameters will be discussed there.

6
Result and Discussion

The results of both travel-time and volume predictions are presented in tables in this chapter. The performances of SVR-predictor combined with three scaling methods, (Robust-scaling, Standard-scaling and Min-Max-scaling) were compared. Moreover, several feature sets were tested.

6.1 Travel-time prediction

As we assumed the travel time of a given route in the morning and afternoon are independent of each other, the same prediction procedure was applied for every route in the morning and afternoon respectively. SVR was used as the main prediction method. After testing with several experiments (with different values chosen randomly), radial basis function (RBF) was chosen as the kernel function, with $\gamma = 0.005$ and $\epsilon = 0.5$. Parameter C was chosen according to

$$max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|) \tag{6.1}$$

where $\bar{y}$ and σ_y are the mean and the standard deviation of the y values of training data [7]. SVR with RBF has been found less sensitive to preprocessing of data such as scaling [8].
Many cross-validation experiments were conducted: using different scaling methods, different amount of training data, and different features sets. Two basic features were always included: time window position and the previous two-hour travel time. *Time window position*: The prediction is for every 20-minute time window of the rush hours (rush hours are defined as 08:00-10:00 and 17:00-19:00), therefore the rush hours are split into six 20-minute time windows. For example, for the rush hours in the morning, 8:00 am-8:20 am ([8:00, 8:20)) is the first position, 8:20 am-8.40 am ([8:20, 8:40))is the second, and so on. *Previous Two-hour travel time*: They are the two-hour travel time data before the rush hours. For instance, the previous two-hour travel time for the rush hours in the morning are the data from 6:00 am to 8:00 am. They are also split into six 20-minute time windows.
Obviously, the travel-time are a result of dynamic interplay of traffic demand and traffic supply [14]. High traffic flow indicates high traffic demand. Factors influencing traffic demand include temporal effects like daily and weekly pattern, as well as holiday [21]. Factors influencing the traffic supply includes crashes, road works, weather, etc. For this reason, extra features were added one by one and the predictive performance of each resulting model was evaluated by comparing the validation and the prediction result. Additional features that can capture the traffic demand

are as follows. *Special days*: working days, weekends, or holidays. *Tollgate volume*: this feature is the volume of the tollgate of the target route. For example, when predicting the travel time of route A-2, the tollgate volume is the volume at tollgate 2 (shown in Fig. 3.1). *Adjacent tollgate volume*: this feature is the volume of the target route's adjacent tollgate. If two routes come from the same intersection and go to different tollgates, one of the two is the target route, and the other is the adjacent route. The tollgate of the adjacent route is called adjacent tollgate. For example, for route A-2, the adjacent tollgate volume is the volume of tollgate 3.
The predictive performances of using SVR combined with different scaling-methods are presented in Table 6.1 and Table 6.2. The results of the experiments using two different amount of training data sets are shown in Table 6.1 (training data from 19/7 to 17/10) and in Table 6.2 (training data from 19/9 to 17/10). The results of the experiments using different sets of features are shown in Table 6.3.

Table 6.1: Average MAPE from 13-fold cross-validation experiments with features: time window position and two-hour travel time; Data used for training are from 19/7 to 17/10; Data used to test are from 18/10 to 24/10;

Scaling method	validation result	prediction of test data
Robust-scaling	0.2302	0.1886
Standard-scaling	0.2296	0.1902
Min-Max-scaling, [0,1]	0.2276	0.1935
No scaling	0.2464	0.2081

Table 6.2: Average MAPE from 4-fold cross-validation experiments with features: time window position and two-hour travel time; Data used for training are from 19/9 to 17/10; Data used to test are from 18/10 to 24/10;

Scaling method	validation result	prediction of test data
Robust-scaling	0.1901	0.2073
Standard-scaling	0.1888	0.2083
Min-Max-scaling, [0,1]	0.1811	0.1928
No scaling	0.1977	0.2001

Comparing Table 6.1 and Table 6.2, one can see that using fewer weeks data for training gives better validation results, but worse prediction results. This also means that our experiments did not show anything conclusive about the influence of season on the travel time prediction (note that the period 19/7 to 18/9 is summer season). Similarly, our experiments (see Table 6.4) suggest that most of the weather-related features did not increase predictive performance of our models. If any, only temperature was worth adding. Based on the experiments with the same amount of training data (data from 19th September to 17th October), adding more features (tollgate volume and adjacent tollgate volume) provides better validation and prediction results (Table 6.3).

Table 6.3: Average MAPE from different cross-validation experiments with Min-Max-scaling in range [0,1] and features: time window position and two-hour travel time; Data used for training are from 19/9 to 17/10; Data used to test are from 18/10 to 24/10;

Extra Feature(s)	validation result	prediction of test data
None	0.1811	0.1928
Special days	0.1795	0.1920
Tollgate volume (vol)	0.1770	0.1931
Tollgate volume & special days	0.1773	0.1938
Tollgate vol. & adjacent tollgate vol.	0.1771	0.1900

The best experimental result from the travel-time prediction task appears in Table 6.1 by applying Robust-scaling with the two basic features (the previous two-hour travel time and time window position). From Table 6.1 and Table 6.2, using scaling method gives better predictive performance compared to no scaling. Robust-scaling seems to be particularly good for time series with more varying patterns (that include summer season), while Min-Max-scaling seems to be particularly good for time series with more similar patterns.

Table 6.4: Average MAPE from different cross-validation experiments with Min-Max-scaling in range [0,1] and features: time window position and two-hour travel time; Data used for training are from 19/7 to 17/10; Data used to test are from 18/10 to 24/10;

Extra Feature	validation result	prediction of test data
None	0.2276	0.1935
pressure	0.2241	0.1934
sea pressure	0.2241	0.1934
wind direction	0.2275	0.1924
wind speed	0.2278	0.1936
temperature	0.2228	0.1894
relative humidity	0.2280	0.1900
precipitation	0.2280	0.1936

6.2 Volume prediction

Similarly to Task 1, in order to build a good model for Task 2, we addressed the following sub-task: given training data for the period of 19th September to 17th October, estimate the average volume for each of the tollgate-direction pairs, per 20 minutes interval, during rush hours (08:00-10:00 and 17:00-19:00) for the period of 18th October to 24th October.
As we assumed the volume of a given tollgate direction pair in the morning and in the afternoon are independent of each other, the same prediction procedure was

applied for all tollgate direction pairs in the morning and afternoon respectively. The average error of all tollgate direction pairs was calculated using MAPE defined in Eq. 5.2. SVR was applied for the volume prediction too. After testing with several experiments (with different values chosen randomly), radial basis function (RBF) was chosen as the kernel function with $\gamma = 0.01$ and $\epsilon = 0.01$. Parameter C was chosen according to Eq. 6.1.
The feature selection strategy for volume prediction was similar as for travel time prediction (Section 6.1). The two basic features here were time window position and the previous two-hour volume. The previous two-hour volume means the two hours volume before the rush hours to be predicted and time window position is similar as in Section 6.1.
The results of performances by using different scaling-methods combined with SVR are presented in Table 6.5. In addition, the comparisons of performances for different features are presented in Table 6.6.
Traffic volume depends on many factors, including time of day, day of week, holiday, weather, etc. For this reason, an additional feature called special days (explained in Section 6.1) to capture the holidays and weekends effect was added. Moreover, other features (basically extracted from the provided volume data), including the number of vehicles with ETC and the number of vehicles have vehicle model n ($n \in [0, 7]$), were also tested in our experiments (see Table 6.6).

Table 6.5: Average MAPE from cross-validation experiments with features: time window position and two-hour volume; Data used for training are from 19/9 to 17/10; Data used to test are from 18/10 to 24/10;

Scaling method	validation result	prediction of test data
Robust-scaling	0.2710	0.1472
Standard-scaling	0.2717	0.1502
Min-Max-scaling, [0,1]	0.3467	0.1526
No scaling	1.0374	0.3128

For the volume prediction, applying SVR combined with a scaling method gives a huge improvement to the result compared with only using SVR, see Table 6.5. And again, it appears that Robust-scaling is particularly good for time series with more varying patterns. Note that the period of 1st October to 7th October is a big holiday period in China and it is widely known that the traffic volume is very different during that period compared to usual days.
The best performance shows up in Table 6.6, with features: two-hour volume, time window position, vehicle model 6, vehicle model 7, and special days. Table 6.6 suggests that the feature special days is a very important feature for traffic volume prediction.

Table 6.6: Average MAPE from cross-validation experiments with Robust-scaling and features: time window position & two-hour volume; Data used for training are from 19/9 to 17/10; Data used to test are from 18/10 to 24/10;

Extra Feature	validation result	prediction of test data
None	0.2710	0.1472
special days	0.2647	0.1470
use ETC	0.3605	0.1705
vehicle model (veh. mod.) 1	0.2854	0.1472
veh. model 2	0.2759	0.1621
veh. model 3	0.3240	0.1531
veh. model 4	0.3138	0.1476
veh. model 5	0.3107	0.1504
veh. model 6	0.2708	0.1476
veh. model 7	0.2738	0.1447
veh. model 7 & special days	0.2682	0.1440
veh. mod. 6 & veh. mod. 7 & special days	0.2691	0.1436

6.3 Generalization

Based on the experiment results from previous sections, we can conclude: 1. SVR with a scaling method performs better compared to without scaling; 2. Robust-scaling is specially good for time series with varying patterns; 3. Min-max-scaling is specially good for time series with similar patterns.
Anyway, this derived conclusion is strongly based on the provided input data. If the input training data is different, can we draw the same conclusion? In other words, we want to generalize the conclusion for different traffic data. However, due to the lack of other traffic data, we analyze the following question instead:

Does the conclusion still hold if other parts of the data had been missing?

If other parts of the data had been missing, we start from some slightly different data. The basic idea to address this question is: randomly delete some values from original data (pretend those values are missing) and run the same experiment for a slightly different input data. In summary, a detailed description of the methodology listed in 4 steps

1. delete $p\%$ values from the original data randomly. We investigate five p values, $p = 10, 20, 30, 40, 50$.
2. using Complementary and linear interpolation to fill in the gaps (missing parts).
3. taking the data obtained in step2, for each scaling method (Robust/Standard/Min-max-scaling), run the experiment(see Section 5.4) with a fix feature set and a fix SVR-setting. For simplicity reason, we use basic feature set(time window position and two-hour travel time), RBF-kernel, and SVR parameter

$\epsilon = 0.5, \gamma = 0.005$. The output from this step is a table similar to Table 6.1, but changed values.

4. repeat step1 to step3 N times ($N >> 0$).

The procedure was repeated 100 times ($N = 100$) and five levels of percentages ($10\%, 20\%, 30\%, 40\%, 50\%$) were investigated. The results are reported in Figure 6.1, 6.2, 6.3, 6.4, 6.5.

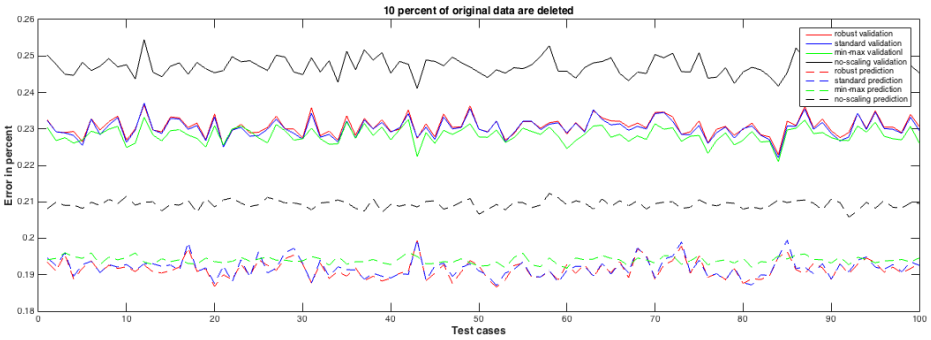

Figure 6.1: Validation and prediction error for 100 experiments with 10% deleted data.

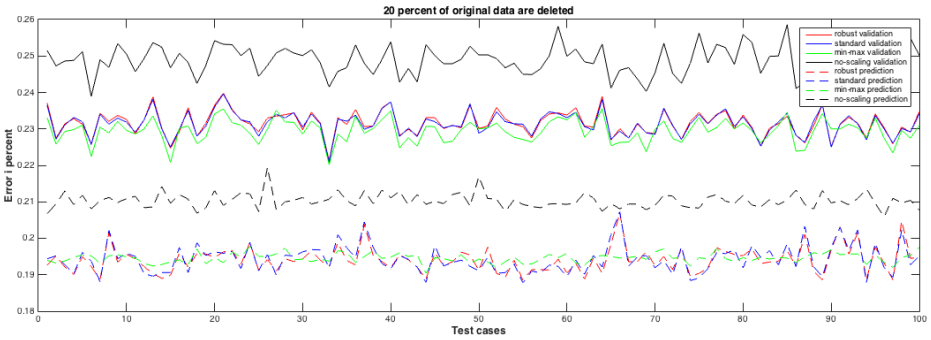

Figure 6.2: Validation and prediction error for 100 experiments with 20% deleted data.

We can see that, the performance of no-scaling (the black lines) is worst among all experiments and the overall performance is improved by using a scaling method. The performances of Robust (red lines) and Standard-scaling (blue lines) are very similar. Compared to the dashed red and dashed blue line, as more values are deleted, the dashed green line (prediction results of Min-max scaling) gradually drop down. This means: as more values are deleted, Min-max scaling gradually perform better than other scaling methods. An explanation is: the more values are deleted, the more outliers disappear and are replaced with smoother values (since we use complementary and linear interpolation to fill in the deleted values). In other words, data after deletion and filling (filling in missing values) become smoother and contain less outliers compared to the original data. Consequently, Min-max scaling is particularly good for time series with similar patterns.

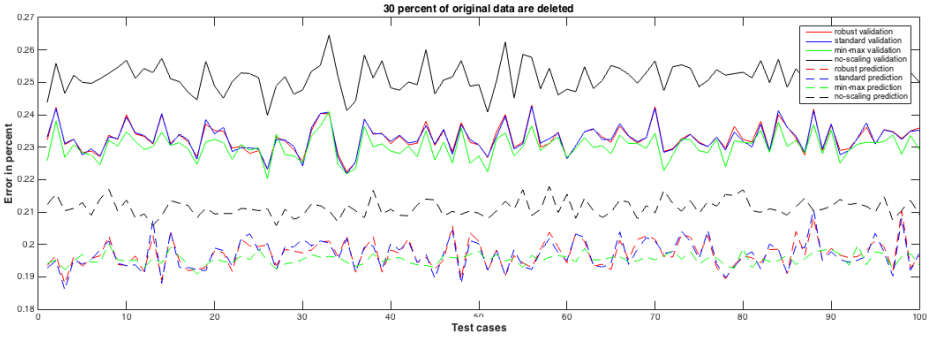

Figure 6.3: Validation and prediction error for 100 experiments with 30% deleted data.

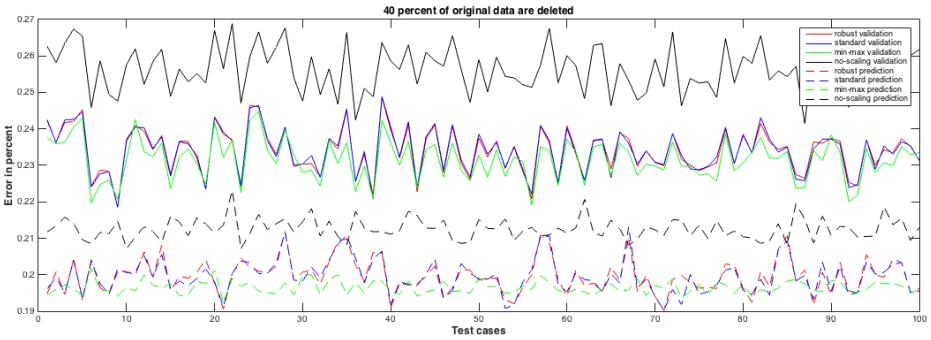

Figure 6.4: Validation and prediction error for 100 experiments with 40% deleted data.

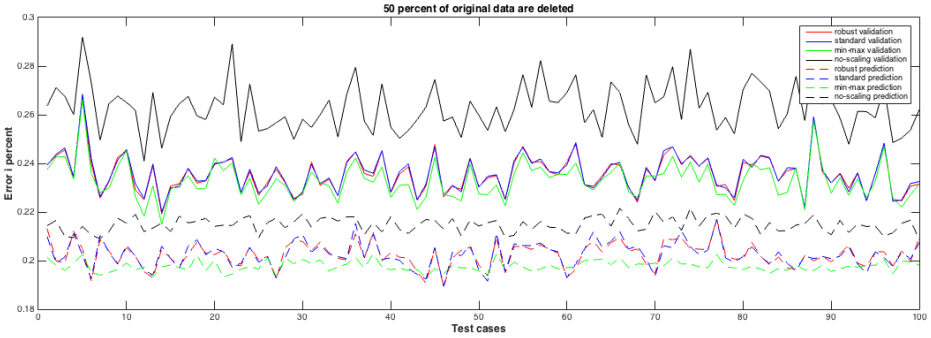

Figure 6.5: Validation and prediction error for 100 experiments with 50% deleted data.

7
Conclusion

7.1 Conclusion

In this experiment, we demonstrate application of SVR for travel-time prediction over a very short distance in rush hours and tollgate traffic volume prediction in rush hours. The performances of SVR-predictor combined with three scaling methods, Robust-scaling, Standard-scaling and Min-Max-scaling were compared.
Our results from travel time and volume predictions (Section 6.1 and Section 6.2) suggested that SVR with a scaling method performs better compared to without scaling, Robust-scaling is particularly good for time series with varying patterns, and Min-Max-scaling is particularly good for time series with more similar patterns. An additional work for generalization of those above suggestions was done in Section 6.3. The results from generalization part confirm that SVR with a scaling method performs better compared to without scaling and Min-Max-scaling is particularly good for time series with more similar patterns. The suggestion about Robust-scaling is particularly good for time series with varying patterns, can not be confirmed or refuted directly in the generalization section.
Features that capture different travel-time/volume influencing factors were analyzed in the experiments. In general, SVR combined with scaling provides a more accurate prediction than without scaling, especially for volume prediction. Adding additional features (travel-time/volume influencing factors) does not give significant improvement.
When our model was applied to Task 1, our travel-time prediction error, around 0.19, differs only 0.02 from the best result obtained by other contestants (this is a competition task, the best prediction result was announced). Similarly, when our model was applied to Task 2, our volume prediction result, around 0.144, differs 0.03 from the best result.
We conclude, for the training data containing many outliers (like holiday data) and without deep analysis of the data (no data pruning), SVR combined with scaling method can still provide reasonable prediction results.

7.2 Future work

In this project, we estimate the missing values by using an own developed method, “Complementary“ with linear interpolation. If other methods are used to estimate the missing values, the results of travel time prediction would be different. For future work, it would be interesting to compare the results if only linear interpolation

or only “Complementary“ is applied. Does the “Complementary“ with linear interpolation perform better than only linear interpolation/“Complementary“? Does the conclusion still hold? Another future work is to apply our model to other similar traffic data for similar tasks and to see the performance.

Bibliography

[1] Kdd2017. https://tianchi.aliyun.com/competition/information.htm?spm=5176.100067.5678.2.ru0ea4&raceId=231597. Accessed June 15, 2017.

[2] Linear interpolation. https://www.encyclopediaofmath.org//index.php?title=Linear_interpolation&oldid=16431. Accessed July 7, 2017.

[3] Preprocessing data. http://scikit-learn.org/stable/modules/preprocessing.html. Accessed June 15, 2017.

[4] Robustscaler. http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html. Accessed June 15, 2017.

[5] Debasish Basak, Srimanta Pal, Dipak Ch, and Ra Patranabis. Support vector regression. In *Neural Information Processing Letters and Reviews*, pages 203–224, 2007.

[6] R. Cayford, W.H. Lin, C. Daganzo, Berkeley. Institute of Transportation Studies University of California, Partners for Advanced Transit, and Highways (Calif.). *The NETCELL Simulation Package: Technical Description*. California PATH research report. California PATH Program, Institute of Transportation Studies, University of California, Berkeley, 1997.

[7] Vladimir Cherkassky and Yunqian Ma. Practical selection of svm parameters and noise estimation for svm regression. *Neural Networks*, 17(1):113 – 126, 2004.

[8] S Crone, J Guajardo, and R Weber. *The impact of preprocessing on support vector regression and neural networks in time series prediction*, pages 37–42. unknown, 2006.

[9] AiLing Ding, XiangMo Zhao, and LiCheng Jiao. Traffic flow time series prediction based on statistics learning theory. In *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems*, pages 727–730, 2002.

[10] Harris Drucker, Chris J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 9*, pages 155–161. MIT Press, 1997.

[11] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.

[12] Piotr Juszczak, David M.J. Tax, and Robert P. W. Duin. Feature scaling in support vector data description. In E.F. Deprettere, A. Belloum, J.W.J. Heijnsdijk, and F. van der Stappen, editors, *Proc. ASCI 2002, 8th Annual Conf. of the Advanced School for Computing and Imaging*, pages 95–102, 2002.

[13] A. Kotsialos, M. Papageorgiou, C. Diakaki, Y. Pavlis, and F. Middelham. Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool metanet. *IEEE Transactions on Intelligent Transportation Systems*, 3(4):282–292, Dec 2002.

[14] J.W.C. Lint. *Reliable travel time prediction for freeways: bridging artificial neural networks and traffic flow theory*. TRAIL Research School, 2004.

[15] Chi-Jie Lu, Tian-Shyug Lee, and Chih-Chou Chiu. Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, 47(2):115 – 125, 2009.

[16] K. R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. *Predicting time series with support vector machines*, pages 999–1004. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[17] Klaus-Robert Müller, Alexander J. Smola, Gunnar Rätsch, Bernhard Schökopf, Jens Kohlmorgen, and Vladimir Vapnik. Advances in kernel methods. chapter Using Support Vector Machines for Time Series Prediction, pages 243–253. MIT Press, Cambridge, MA, USA, 1999.

[18] Simon Oh, Young-Ji Byon, Kitae Jang, and Hwasoo Yeo. Short-term travel-time prediction on highway: A review of the data-driven approach. *Transport Reviews*, 35(1):4–32, 2015.

[19] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August 2004.

[20] J.W.C. van Lint, S.P. Hoogendoorn, and H.J. van Zuylen. Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C: Emerging Technologies*, 13(5):347 – 369, 2005.

[21] Chun-Hsin Wu, Jan-Ming Ho, and D. T. Lee. Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):276–281, Dec 2004.

[22] QI Yang and Haris N. Koutsopoulos. A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Transportation Research Part C: Emerging Technologies*, 4(3):113 – 129, 1996.

[23] Chi-Yuan Yeh, Chi-Wei Huang, and Shie-Jue Lee. A multiple-kernel support vector regression approach for stock market price forecasting. *Expert Systems with Applications*, 38(3):2177 – 2186, 2011.

[24] Pao-Shan Yu, Shien-Tsung Chen, and I-Fan Chang. Support vector regression for real-time flood stage forecasting. *Journal of Hydrology*, 328(3):704 – 716, 2006. The ICWRER - Symposium in Dresden, Germany.

A

Data illustration

Figures A.1-A.11 show 20-minute travel time at all routes in the morning and afternoon during the period 19th September to 24th October. The data used in Figures A.1-A.11 have been filled in by "Complementary" and linear interpolation (Section 5.1.2). Figures A.12-A.20 show 20-minute volume at all tollgates directions in the morning and afternoon during the same period.

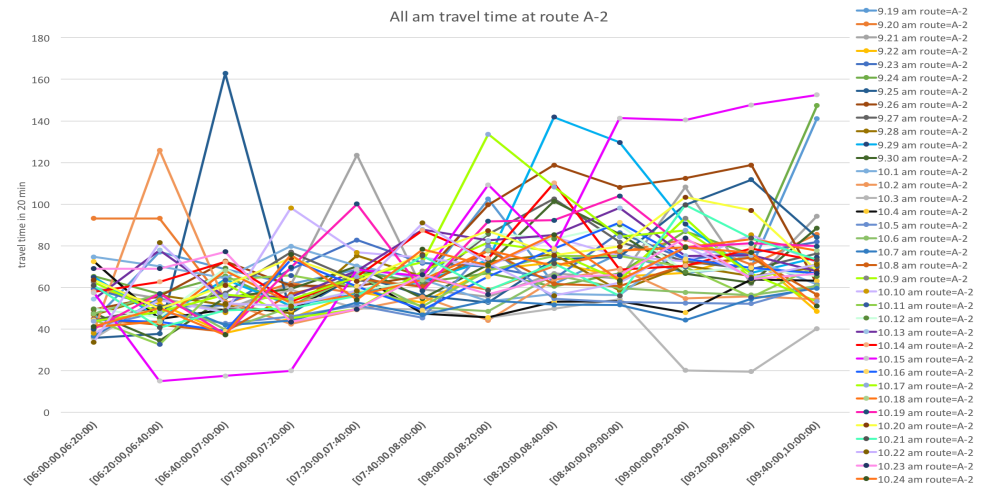

Figure A.1: 20-minute travel time at route A-2 in the morning

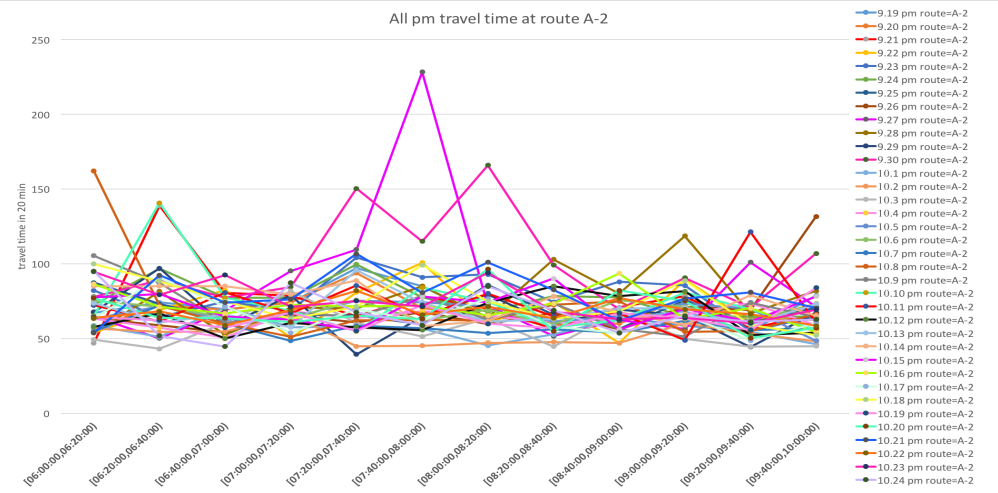

Figure A.2: 20-minute travel time at route A-2 in the afternoon

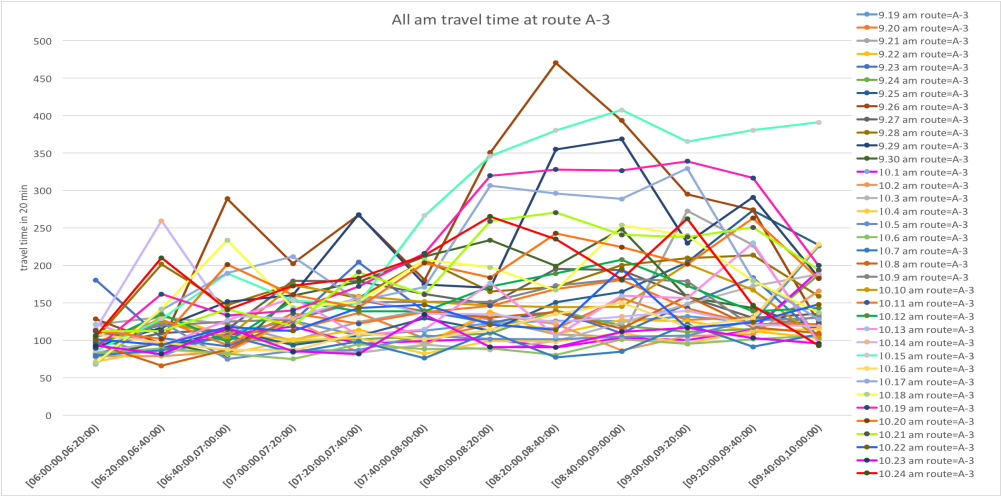

Figure A.3: 20-minute travel time at route A-3 in the morning

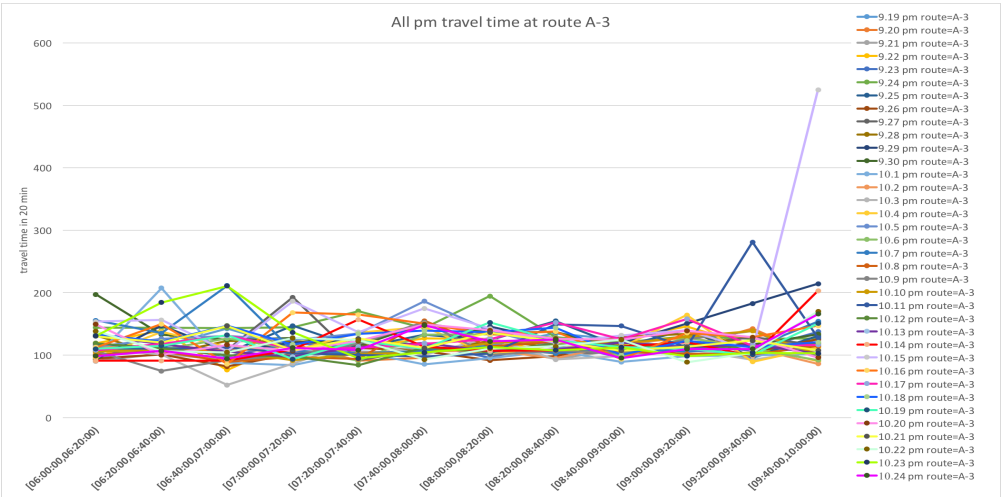

Figure A.4: 20-minute travel time at route A-3 in the afternoon

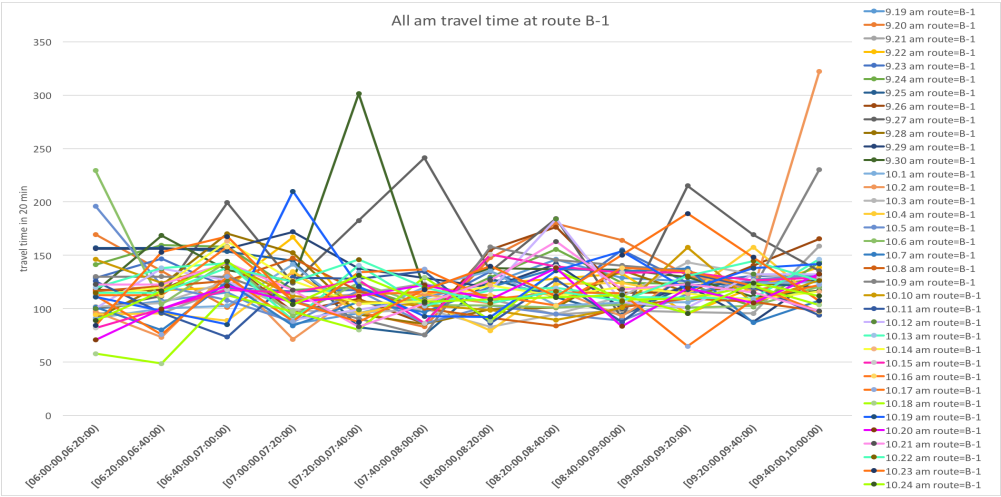

Figure A.5: 20-minute travel time at route B-1 in the morning

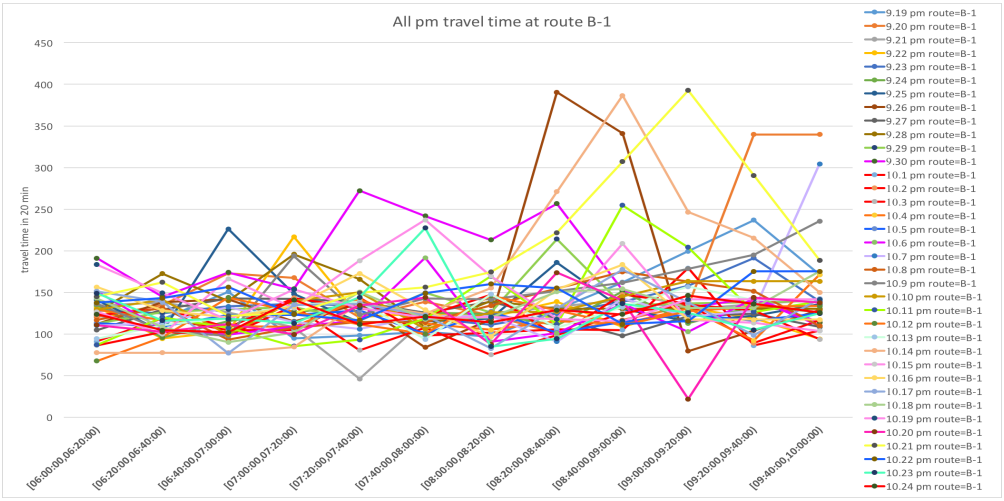

Figure A.6: 20-minute travel time at route B-1 in the afternoon

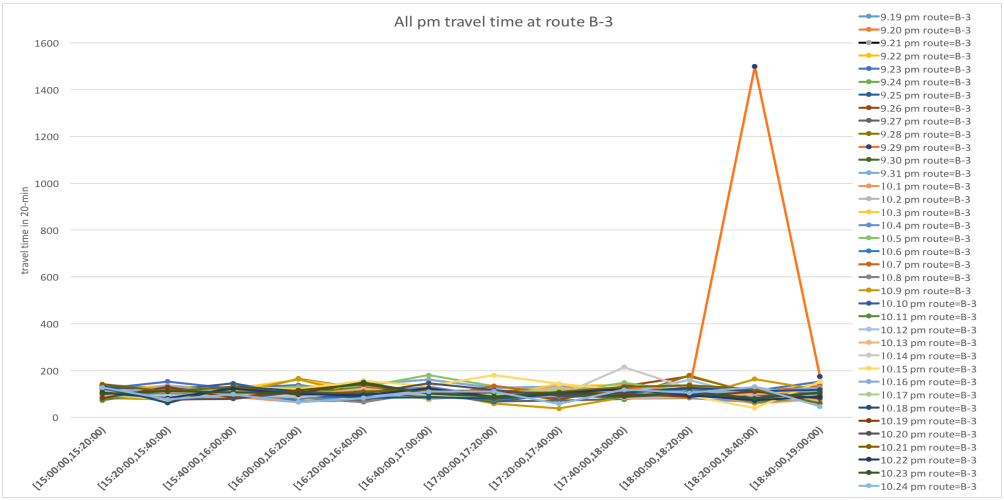

Figure A.7: 20-minute travel time at route B-3 in the morning

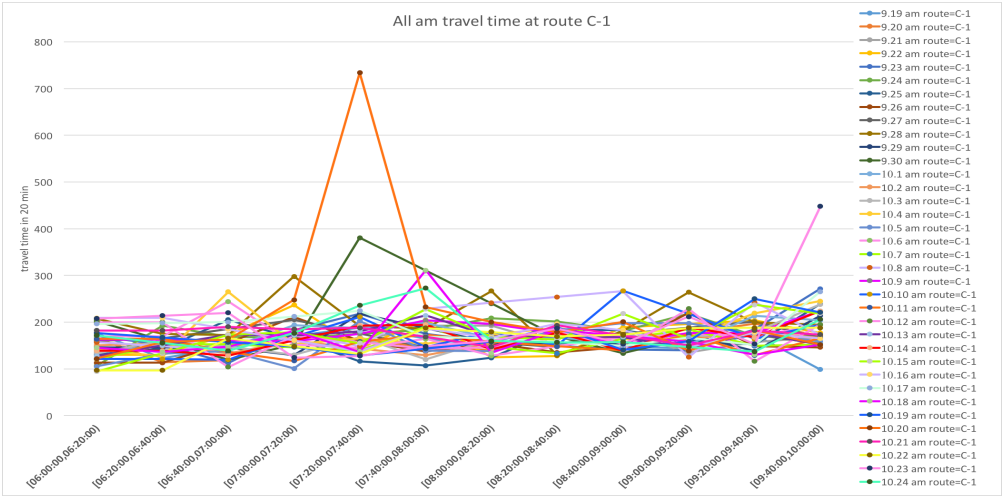

Figure A.8: 20-minute travel time at route C-1 in the morning

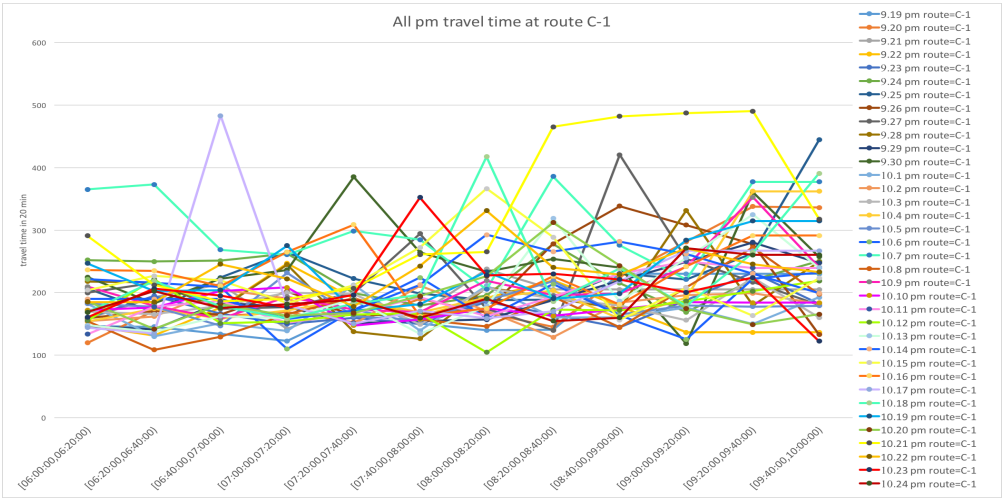

Figure A.9: 20-minute travel time at route C-1 in the afternoon

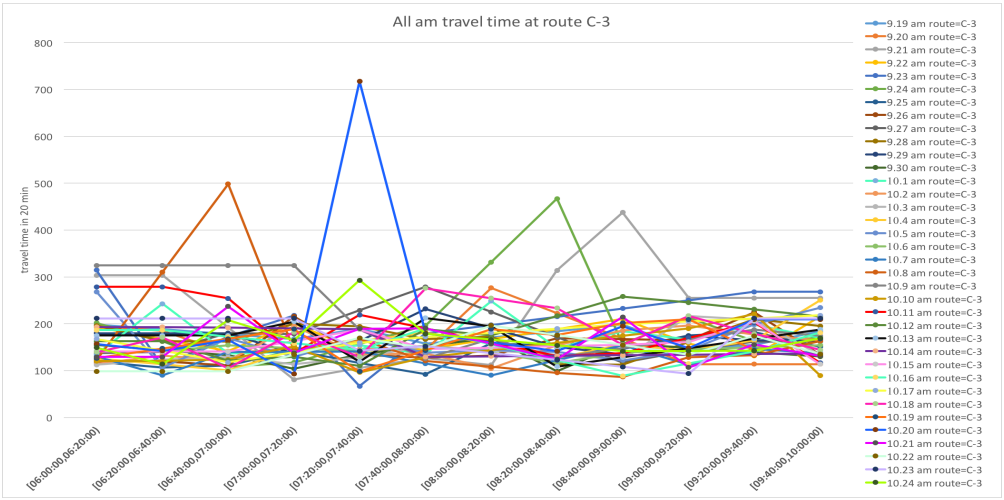

Figure A.10: 20-minute travel time at route C-3 in the morning

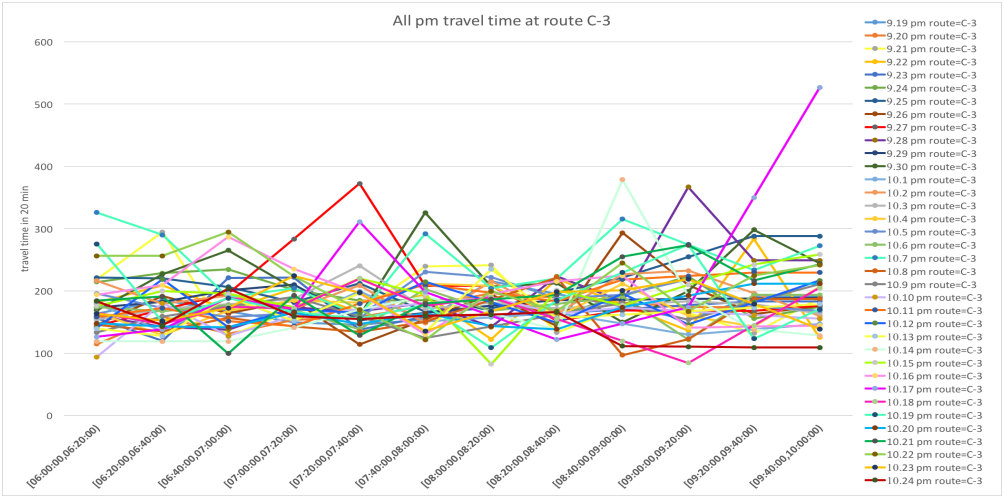

Figure A.11: 20-minute travel time at route C-3 in the afternoon

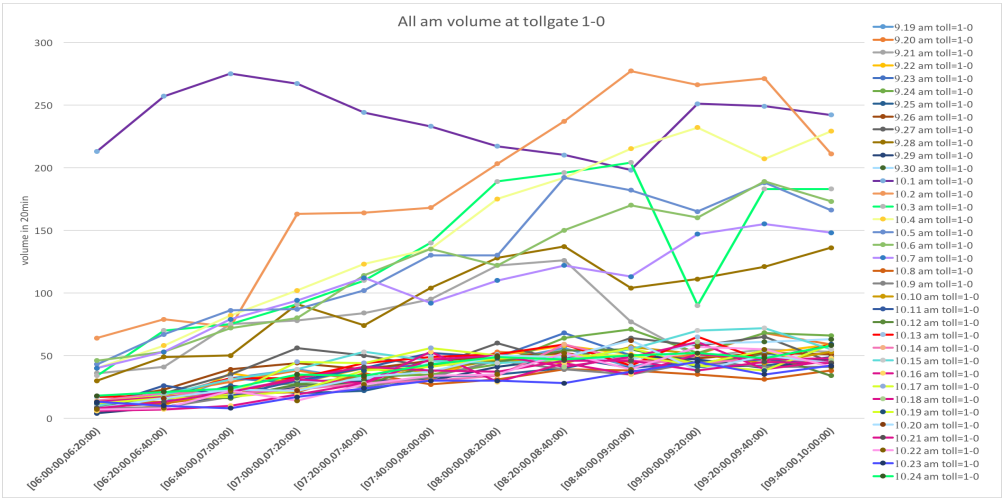

Figure A.12: 20-minute volume at tollgate 1-0 in the morning

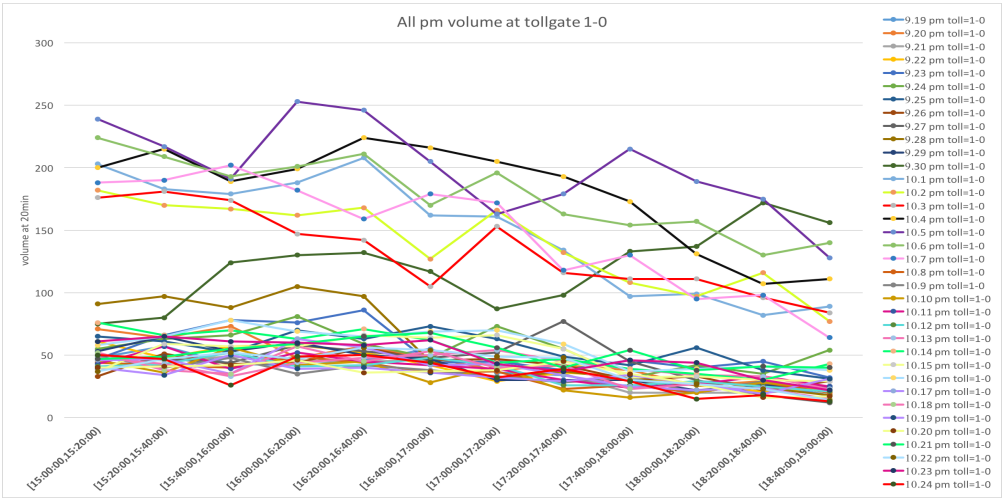

Figure A.13: 20-minute volume at tollgate 1-0 in the afternoon

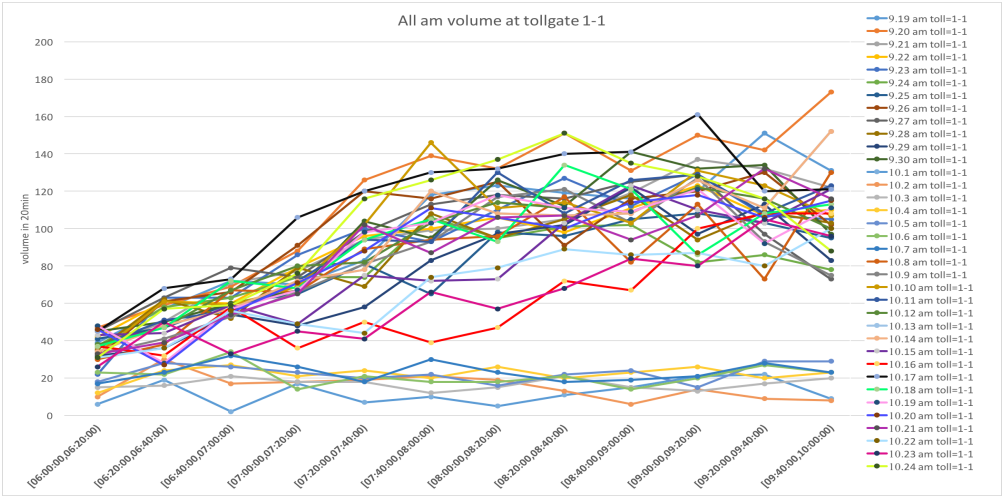

Figure A.14: 20-minute volume at tollgate 1-1 in the morning

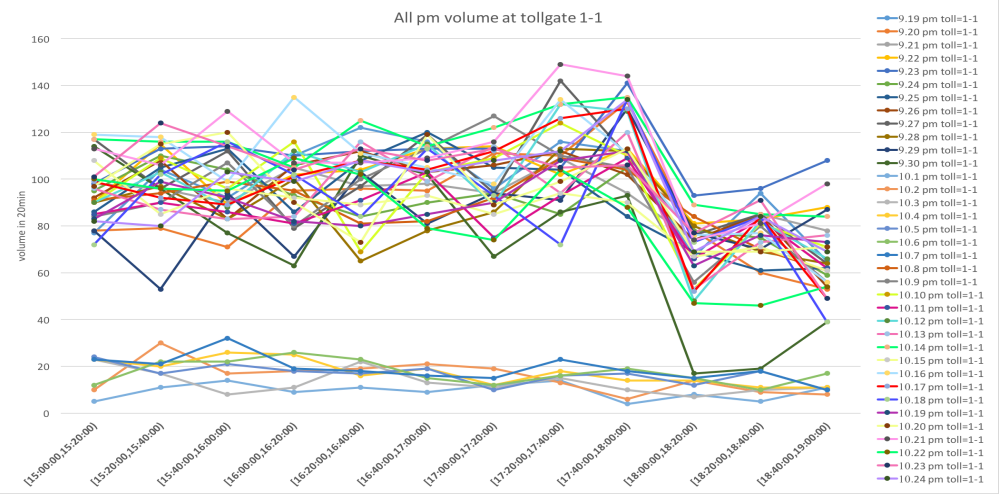

Figure A.15: 20-minute volume at tollgate 1-1 in the afternoon

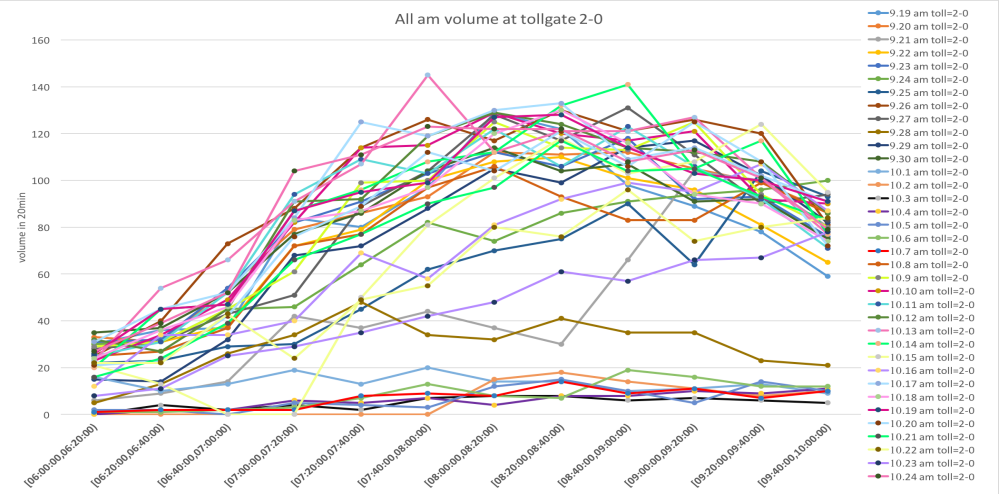

Figure A.16: 20-minute volume at tollgate 2-0 in the morning

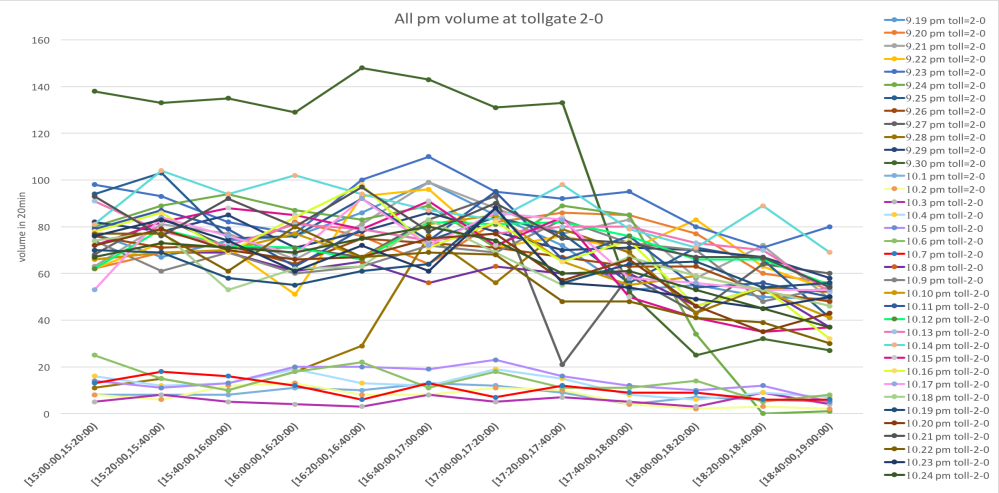

Figure A.17: 20-minute volume at tollgate 2-0 in the afternoon

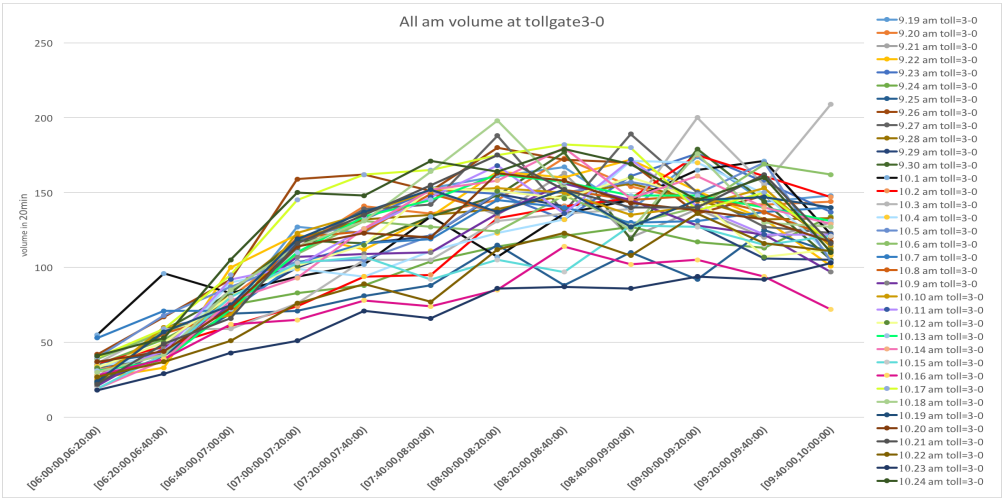

Figure A.18: 20-minute volume at tollgate 3-0 in the morning

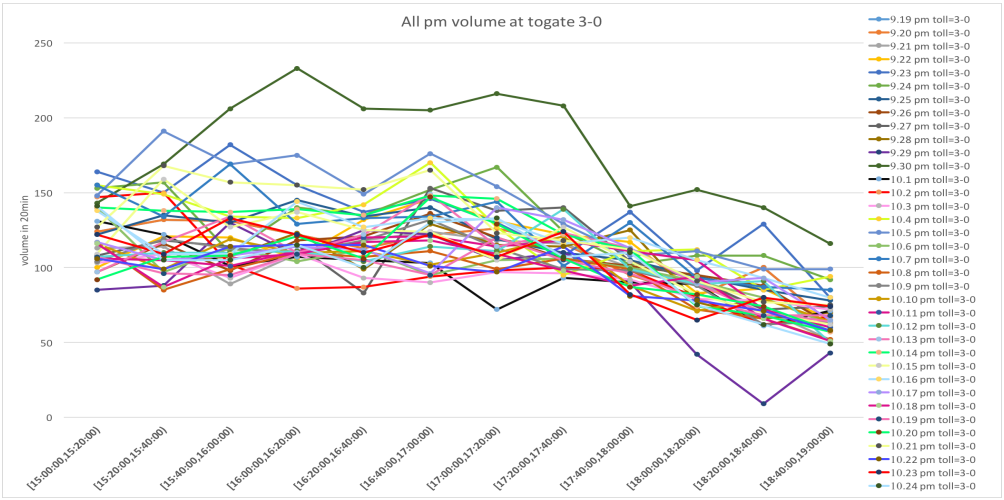

Figure A.19: 20-minute volume at tollgate 3-0 in the afternoon

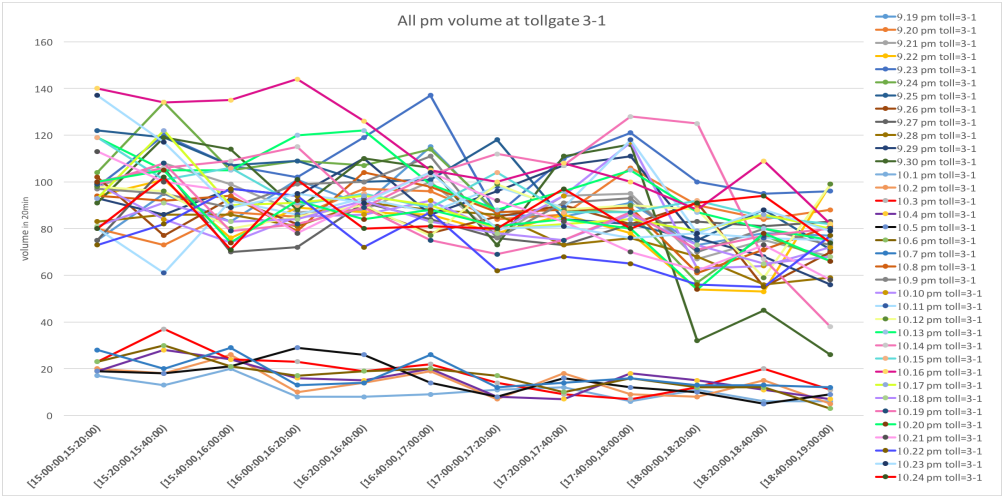

Figure A.20: 20-minute volume at tollgate 3-1 in the afternoon