Car-Following Described by Blending Data-Driven and Analytical Models: A Gaussian Process Regression Approach

Soldevila, Ignasi Echaniz; Knoop, Victor L.; Hoogendoorn, Serge

DOI
10.1177/03611981211032648

Publication date
2021

Document Version
Final published version

Published in
Transportation Research Record

Citation (APA)
Soldevila, I. E., Knoop, V. L., & Hoogendoorn, S. (2021). Car-Following Described by Blending Data-Driven and Analytical Models: A Gaussian Process Regression Approach. *Transportation Research Record*, *2675*(12), 1202-1213. https://doi.org/10.1177/03611981211032648

TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Research Article

Transportation Research Record
1–12

DOI: 10.1177/03611981211032648
journals.sagepub.com/home/trr
SAGE

Car-Following Described by Blending Data-Driven and Analytical Models: A Gaussian Process Regression Approach

Ignasi Echaniz Soldevila[1], Victor L. Knoop[2], and Serge Hoogendoorn[2]

Abstract

Traffic engineers rely on microscopic traffic models to design, plan, and operate a wide range of traffic applications. Recently, large data sets, yet incomplete and from small space regions, are becoming available thanks to technology improvements and governmental efforts. With this study we aim to gain new empirical insights into longitudinal driving behavior and to formulate a model which can benefit from these new challenging data sources. This paper proposes an application of an existing formulation, Gaussian process regression (GPR), to describe individual longitudinal driving behavior of drivers. The method integrates a parametric and a non-parametric mathematical formulation. The model predicts individual driver's acceleration given a set of variables. It uses the GPR to make predictions when there exists correlation between new input and the training data set. The data-driven model benefits from a large training data set to capture all driver longitudinal behavior, which would be difficult to fit in fixed parametric equation(s). The methodology allows us to train models with new variables without the need of altering the model formulation. And importantly, the model also uses existing traditional parametric car-following models to predict acceleration when no similar situations are found in the training data set. A case study using radar data in an urban environment shows that a hybrid model performs better than parametric model alone and suggests that traffic light status over time influences drivers' acceleration. This methodology can help engineers to use large data sets and to find new variables to describe traffic behavior.

Since the introduction of powerful micro-simulation tools in the last decades of the 20th century, the way that experts approach traffic modeling has changed. Fast computers have made it possible to use advanced traffic micro-simulation software packages which describe the individual agent's driving behavior. Today, the number of traffic microscopic simulation models is vast and the simulation approaches and model applications are diverse. Traffic engineers rely on microscopic traffic software to examine signalized roundabouts, optimize signalized intersections, to test a wide range of traffic management measures such as ramp metering or high-occupancy lanes measures, to estimate individuals vehicle traffic emissions, and to design and test control algorithms for autonomous vehicles. Individual driver behavior in microscopic traffic models is based on a combination of mainly three models: car-following (CF) models, lane-changing models, and gap acceptance models. CF models are the sub-models that describe the interactions with preceding vehicles in the same lane (*1*). Lots of research has been carried out on this topic, from the Gazis–Herman–Rothery model at the General Motors research labs in the fifties and earlier sixties to modern models such as the Intelligent Driver Model in the current century (*2*). Good examples of literature review on historical CF models can be found in Brackstone and McDonald (*2*) and Treiber and Kesting (*3*).

During the past decades, existing parametric CF models have been enumerated and calibrated using traditional techniques and small yet accurate data sets. Traditional optimization calibration techniques simply consist of maximizing the fit of a particular parametric equation to the data, given a set of parameters and an objective function. During recent years, large data sets are becoming available thanks to technology improvements and governmental efforts such as the Next

[1]InControl Simulation Software, Utrecht, The Netherlands
[2]Delft University of Technology, Delft, The Netherlands

Corresponding Author:
Ignasi Echaniz Soldevila, ignasi.echaniz@incontrolsim.com

Generation Simulation data sets (*4*). Contrary to the traditional data sets, these data sets contain large amount of data, for example, thousands of trajectories. Nonetheless, they may have errors, noise, and could only represent small data space regions. Calibration of parametric models using these large data sets can encounter several difficulties. First, large amounts of data represent a challenge to traditional optimization schemes because of computational time issues. Second, parametric traffic models will always be constrained by their specific model parameters. A lot of work has been done already in relation to model calibration and sensitivity analysis on parameters, such as Daamen et al. (*5*). Moreover, studies such as Ossen (*6*) have shown that drivers might behave significantly differently in identical situations. Therefore, it is challenging to shape driver behavior in a fixed constrained parametric model. Finally, traditional parametric models might suffer from overfitting or wrong estimations outside calibration data space regions.

Alternatively to parametric formulation, non-parametric models derived from data-driven techniques are rapidly becoming popular; an overview is presented in the next section. The main benefit of such models is that any kind of data can be easily used to try to predict the behavior of drivers, without the need of a specific model enumeration. At the same time, a weakness of data-driven models is that data for all driving conditions in all driving conditions are necessary, which is often lacking for rare situations. To benefit from data-driven formulation and at the same time be able to predict unknown situations, we propose an application of an existing formulation, Gaussian process regression (GPR) with a fixed deterministic mean, to enumerate a CF model. The proposed model relies on a non-parametric data-driven formulation when there are historical data. Its non-parametric formulation gives freedom to the model to capture drivers' behavior, and at the same time, it enhances fast optimization calculations. The model also uses parametric formulation for data regions outside the historical data. Combining both parametric and non-parametric formulation allows us to make adequate predictions inside and outside training data set regions.

Thus, the aim of the paper is twofold:

I. Explore whether GPR formulation can be applied to describe longitudinal behavior of cars;
II. Explore new variables that could help to describe the longitudinal behavior of cars. By applying GPR, we get the chance of getting insights in new variables relationships to describe this behavior. The formulation allows us to easily add or subtract variables to check its relevancy.

To the best of our knowledge, it has not being explored if GPR formulation can be used to elaborate a CF model. Moreover, traffic engineers have not traditionally explored variables such as the traffic light or the distance to the traffic light to describe the behavior of drivers at signalized intersections. Reasons for this are the lack of data, added complexity to (parametric) formulation, or difficult transition of the formulation between environments with and without traffic lights. Overall, using a large data set from new traffic data-collection techniques with non-parametric model formulation is challenging and represents a new line of research in the traffic field. This is elaborated further in the next section. After that, the paper is set up as follows. The following two sections illustrate the problem definition that has inspired this study and propose a methodology to build hybrid GPR models. Then, the paper explains the set up of the case study. Finally, the paper depicts the simulation results and draws the conclusion driven from this study.

Literature Review on Data-Driven Techniques in CF Models

Data-driven techniques such as machine learning, deep learning, or data mining are becoming more popular in the transportation field as they can benefit from a large amount of information. Academic research in machine learning techniques has mainly focused on traffic applications. For instance, Karlaftis and Vlahogianni provide a list of studies using machine learning techniques in traffic operations (traffic forecasting, incident detection, etc.), planning (route and mode choice, analysis of travel behavior, trip generation, etc.), and safety and human behavior (accidents analysis, fitness to drive,,etc.) (*7*). Hofleitner et al. propose a hybrid approach of traditional flow modeling techniques and machine learning to forecast urban travel time with streaming GPS probed data (*8*). Elfar et al. explore the use of three machine learning techniques—logistic regression, random forests, and neural networks—for short-term traffic congestion prediction using vehicle trajectories available through connected vehicles technology (*9*). Lv et al. describe a deep learning approach to predict traffic flows (*10*).

The potential use of data-driven techniques in microscopic traffic modeling and in particular CF models is starting to be widely adopted to predict the longitudinal movement of automated vehicles. Artificial neural network (ANN) was one of the first machine learning techniques to be studied. For example, Panwai and Dia describe a CF model using reactive agent techniques based on a neural network approach for mapping perceptions to actions (*11*). It classifies five driver modes (e.g.,

free driving, following, danger, etc.) with ANN according to speed difference and spacing inputs, to later on apply response rules according to the driver mode. Khodayari et al. describe a complete ANN model given four inputs (spacing, speed difference, speed, and reaction time), one output (acceleration), and only one hidden layer (*12*). Unlike other models, the reaction time is not considered fixed and it is linearly dependent on the spacing and the current acceleration of the driver. Recently, deep learning and reinforcement learning are new lines of research. Wang et al. detail a new deep neural network rather than conventional neural networks to establish a CF model (*13*). It uses speed, speed difference, and position difference observed in few time intervals as inputs, and it is built in a data-driven way. Zhu et al. (*14*) use the same explanatory variables as Wang et al. (*13*) and deep learning to enumerate a framework for human-like autonomous vehicle CF planning. The model is trained from trial-and-error interactions based on a reward function. Similarly, Gao et al. establish a reward function for each driver data based on the inverse reinforcement learning algorithm to model complex traffic conditions (*15*). Lee et al. combine a stochastic CF models and deep learning architecture to determine if a driver attempts lane changing in a multi-lane freeway (*16*).

Few studies attempt to build a hybrid model combining parametric and non-parametric formulation. Yang et al. try to improve the machine learning-based CF models by combining them with a kinematics-based CF model (i.e., parametric model) (*17*). It uses machine learning-based models such as Back-Propagation Neural Networks and Random Forest models together with the well-known Gipps model (parametric model). The study shows that both machine learning CF models have better performance when are combined with a parametric model than alone. This proves that combining both parametric (kinematic models) and non-parametric models (data-driven models) is an interesting line of research. To the best of our knowledge there are no studies aiming to use a GPR approach to derive CF models. This is therefore addressed in the current paper.

Problem Definition

CF models aim to simulate the longitudinal driving agent behavior along the road. Existing models aim to mimic the acceleration of a driver, that is, response variable, in time step t, based on a set of predictor variables at time step t-1. As in many other CF studies, this study suggests to predict acceleration of a vehicle n at time step t, that is, a_{n_t}. The proposed predictor variables to describe acceleration are listed as follows:

$v_{n_{t-1}}$	The own speed of vehicle n at time step t-1
$s_{(n,\ n-1)_{t-1}}$	The spacing distance between vehicle n and its leader vehicle n-1 at time step t-1
$\Delta v_{(n,\ n-1)_{t-1}}$	The speed difference between vehicle n and its leader vehicle n-1 at time step t-1
$x^{n_{t-1}}$	The distance of vehicle n to the downstream traffic light at time step t-1
STAT^{t-1}	The status of the downstream traffic light of vehicle n at time step t-1

Figure 1 depicts the response and predictor variable proposed in this project to describe drivers behavior.

The Proposed Methodology

Large data sets brings the opportunity to investigate new data-driven oriented forms of deriving CF models to capture driving behavior, which may be difficult to describe in fixed parametric equations. At the same time, the CF model should be able to predict the acceleration of the driver in any driving situation, even if this situation is outside the training data set. Therefore, we propose a CF model based on GPR with a fixed parametric basis function. The GPR model formulation benefits from the non-parametric formulation in those space regions where there is historical data. Therefore, the model relies on the data correlation instead of on fixed underlying equations if there is a correlation with the training data set. By doing so, we aim to gain new empirical insights into longitudinal driving behavior by capturing all attributes in this data-driven process. The GPR formulation gives flexibility to include and exclude new variables and study their relationships. Finally, the GPR model formulation shifts to an existing parametric model in those space regions with no correlation with the training data set. Model predictions made by the exiting parametric model are not assessed in this paper. This is because the parametric formulation used has not been validated inside or outside space regions of the available data. However, the proposed formulation can be used when there is already a validated parametric formulation in outside space regions of the data. In any case, we ensure the

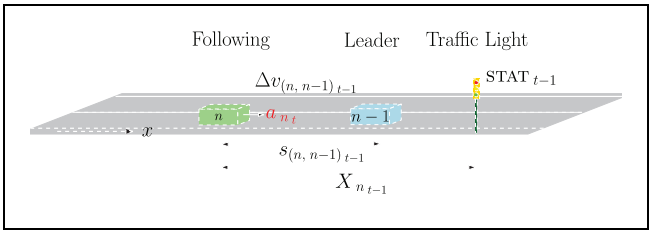

Figure 1. Proposed response and predictor variables to derive a CF model.

completeness of the model. The next sections describe the formulation of the model and the proposed methodology to train CF models.

Gaussian Process Regression

A Gaussian process (GP) is a type of continuous stochastic process which defines a probability distribution for functions (*18*). Consider a training set: $\{(x_i, y_i); i = 1, 2, ..., n\}$, where $\vec{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, drawn from an unknown distribution. A GPR model, it addresses the question of predicting the value of a response variable y_*, given a new input vector x_* and the training data set. Also known as kriging, GPR it is a well-known application enumerated by the French mathematician Georges Matheron and the Russian meteorologist L. S. Gandin in the beginning of the 1960s, based on previous work done of Danie G. Krige (*19*). GPR has been widely used in fields such as mining, meteorology, and statistics among others to optimally predict in space, using observations taken at known nearby locations.

The contribution of this paper is to apply this existing method, that is, GPR, to enumerate a CF model. Therefore, $\vec{x}_i$ contains the predictor variables such as speed and spacing in the training data set and y_i represents the observed acceleration. When a new input vector is given x_*, the model calculates a predicted acceleration y_*.

Now suppose that we pick a particular finite subset of set of random variables indexed by a continuous variable: $f(x), f = \{f_1, f_2, ..., f_n\}$, with indices x_i. In a GP, any such set of random function variables are distributed multivariate Gaussian (*20*):

$$P(f|X) \sim \mathcal{N}(m, K) \tag{1}$$

where $m : \mathcal{X} \rightarrow \mathbb{R}$ is the **mean function** and $K : \mathcal{X}^2 \rightarrow \mathbb{R}$ is the **co-variance function** of a real process $f(x)$:

$$m(x) = \mathbb{E}[f(x)] \tag{2}$$

$$K(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))] \tag{3}$$

Thus, a GPR is completely specified by its mean function and co-variance. The latter can be defined by various kernel functions. The kernel co-variance function describes how far apart the given training data points from each other, that is, correlated between each other. A mathematical description of the GPR and kernel functions can be found in Rasmussen and Williams (*21*).

Training observations typically incorporate noise ($y = f(X) + \varepsilon$). Assuming additive independent identically distributed Gaussian noise ε with σ_n^2, the prior on the noisy observations becomes:

$$cov(y) = K(X, X) + \sigma_n^2 I \tag{4}$$

Usually, the mean function $m(x)$ is assumed to be zero. However, we suggest using a deterministic (fixed) mean function $\mu(x)$. There exist several reasons to include a mean function into the GPR formulation such as including interpretability of the model or convenience of expressing prior information, among others (*21*). In our case, we aim that the basis function is able to predict all those situations lacking in the training data.

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')) \tag{5}$$

Thus, assuming the training set, (X, y), with additive independent identically distributed Gaussian noise ε with variance σ_n^2, a new input point x_* and the desired y_*, and the deterministic fixed mean basis function μ, the covariance function and the joint distribution of the observed target values and the function values at the test locations under the prior are:

$$P(y_*|y, X, x_*) \sim \mathcal{N}(y_*|\bar{f}_*, cov(f_*)) \tag{6}$$

where the predictive mean is

$$\bar{f}_* = \mu(x_*) + K(x_*, X) \underbrace{\left[K(X, X) + \sigma_n^2 I\right]^{-1}(y - \mu(X))}_{\alpha} \tag{7}$$

and the predictive variance is

$$cov(f_*) = K(x_*, x_*) - K(x_*, X)\left[K(X, X) + \sigma_n^2 I\right]^{-1} K(X, x_*) \tag{8}$$

Note that when x_* is uncorrelated with X, the predictive mean μ_* tends to zero as $K(x_*^T, X)$ is small. Thus, the resultant mean function of the predictive distribution is the underlying mean function of the explicit basis function evaluated in the new input $m(x_*)$.

The GPR models need to be trained. This means finding the best parameter values that best fit the training data. The parameters to be optimized, commonly called hyper-parameters, are:

- θ set of parameters from the kernel co-variance function $K(X, X)$.
- σ^2 variance noise of the training data.

To get insights into what does GPR model looks like, a theoretical example is illustrated in Figure 2. Imagine that we train a GPR with a data set (see Figure 2*a*). Then, we use the GPR trained to make prediction using a new data input point x_*. Note that for this single data point input, the GPR prediction is a Gaussian normal distribution with mean m_* and predictive variance as depicted in Equation 8. Consequently, the blue line in Figure 2*a* represents the predictive mean function evaluated to all x axis. As can be

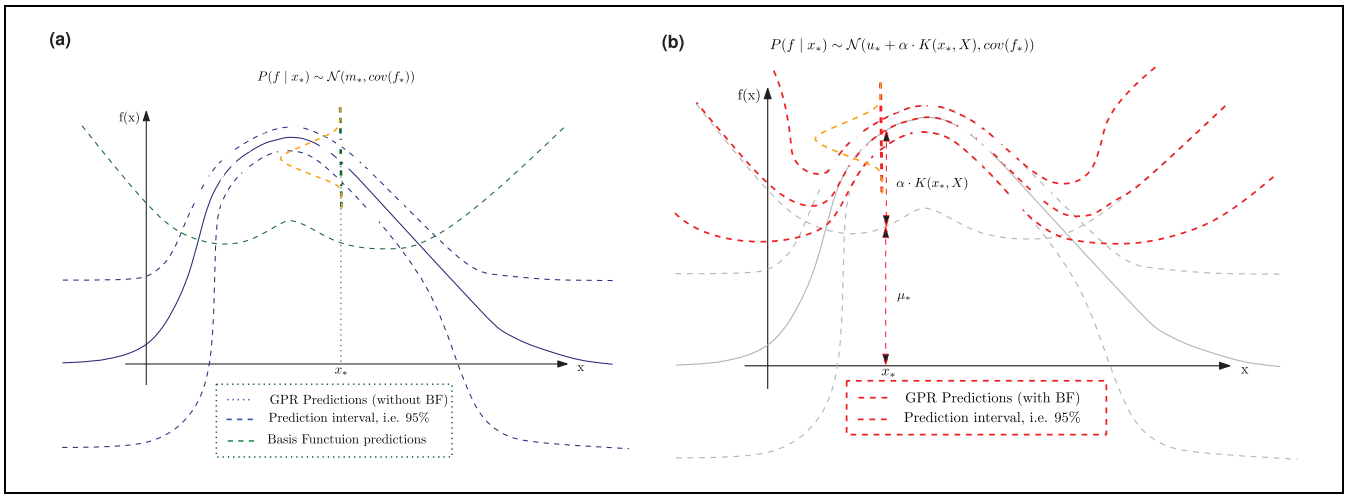

Figure 2. Theoretical interpretation of Gaussian processes regression (GPR) with basis function. (*a*) GPR with zero mean. (*b*) GPR with basis function.

observed in the same figure, the predicted mean tends to the data when observations are close and to zero where there are no data nearby. Furthermore, the variance is small when the given input is close to training data points. The kernel co-variance function evaluated into the new input data point x_*, that is, $K(x_*, X)$ describes how the new data point is correlated to the training data. The variance noise of the training data, that is, σ^2, also plays a major role. It highly affects the total variance of the GPR prediction, that is, the width of the prediction confidence interval, dashed line. The next step is to incorporate a μ basis function as mean in the GPR. By doing so, the GPR relies on data points in those space regions correlated with the training data, and it relies on a specific basis parametric function where new input data points are uncorrelated with the training set (see Figure 2*b*). This is possible by incorporating a basis function as a mean function of the GPR. Then, when making a prediction of a new data point, that is, x_*, the predictive mean is the sum of the evaluation of the new point in the basis function ($\mu(x_*)$) plus the GPR term ($K(x_*, X)\alpha$), which depends on how this point is correlated with the training set. If $K(x_*, X) = 0$, no correlation, the predicted mean will be the basis function. If $K(x_*, X) > 0$, correlation, the predicted mean will deviate from the basis function according to the kernel covariance function. There are two main key points when a basis function is incorporated: (1) the transition between data points and basis function, and (2) how far apart are those two functions originally.

Implementation

We propose the following scheme depicted in Algorithm 1 to derive the optimal hyper-parameters, (θ, σ^2). This approach can be seen as a traditional optimization using GPR formulation.

The following points provide a description of the elements used in the optimization of the hyper-parameters to build the GPR models:

- The training subset consists of the response variable measurements (i.e., $\vec{A}$ acceleration measurements) and the explanatory variable measurements (i.e., $\vec{y} = (\vec{v}, \vec{s}, \vec{\Delta s}, \vec{X}, \hat{STAT})$ speed, spacing, speed difference, distance to the traffic light, and status of the traffic light).
- Initial hyper-parameters are:
  - Kernel parameters in the kernel function, $\theta_0 = (\sigma_l, \sigma_f)$ are set to $mean(std(predictors))$ and $std(response)/\sqrt{2}$ respectively. We use the squared exponential kernel.
  - Initial variance noise of the training data is assumed $\sigma_0^2 = [std(response)/\sqrt{2}]$.

The optimal velocity model (OVM) is chosen as a fixed deterministic basis function. This time-continuous model describes acceleration of a driver based on its own speed and the spacing with its predecessor. Originally introduced by Bando et al. (*22*), the model adapts the actual speed v, to the optimal velocity v_{opt} on a time scale given by the adaptation time τ. The optimal speed v_{opt} increases while spacing increases until the desired speed v_0 is reached on a certain spacing and afterwards it keeps constant:

$$\dot{v}_{\text{OVM}}(s, v) = \frac{(v_{\text{opt}}(s) - v)}{\tau} \tag{9}$$

where

Algorithm 1 Hyper-parameters optimization

Input: Training Subset $(\hat{X}, \hat{y})$, (θ_0, σ_0^2) (Initial hyper-parameters) (μ) Basis Function
1: **Objective Function:** $\hat{\theta}, \hat{\sigma}^2 = \underset{\theta,\sigma}{\operatorname{argmin}} \ \mathscr{F}_{\text{mix}}[s^{\text{sim}}](\theta, \sigma^2)$
2: Build the GPR with the training data- Equation 5: Calculation of $\alpha = \left[K(X,X) + \sigma_n^2 I\right]^{-1}(y - \mu(X))$
3: Prediction of following trajectories. Input: the first point of each trajectory and the trajectory of the leader (from the validation data set). Trajectories are calculated from the mean prediction of acceleration: $E(y_*|y, X, x_*, \theta, \sigma^2, \mu) = \mu(x_*) + K(x_*, X)\alpha$
4: Calculation of $\mathscr{F}_{\text{mix}}$ comparing each simulated trajectory to the observed trajectories in the validation training set
5: Average $\mathscr{F}_{\text{mix}}$
Output: (θ, σ^2) (Hyper-parameters)

$$v_{\text{opt}}(s) = v_0 \frac{\tanh(\frac{s}{\Delta s} - \gamma) + \tanh\gamma}{1 + \tanh\gamma} \tag{10}$$

All OVM parameters τ, v_0, Δs and γ are defined by positive values. OVM parameters values have been set according to Treiber and Kesting (*3*). The two main reasons to choose this model among other linear or nonlinear parametric CF models are:

(i) OVM predicts the acceleration of a driver. This is the same response variable as in this study.

(ii) OVM is one of the simplest CF models. It only uses two explanatory variables (i.e., speed and spacing). The mathematical formulation of the GPR shows that the variables included in the basis function must be also included as a predictor variable in the GPR. This means that by using the OVM as basis function, all GPR models should at least have speed and spacing as predictor variables. Speed and spacing are two main variables in any CF model. Thus, it is not a significant drawback.

Case Study

Training Data Set

We used data collected by roadside units radars at signalized intersections in Amsterdam (NL). The radars were used in one of the first worldwide large-scale field operational tests testing coordinate network-wide deployment of traffic management in practice (*23*). Radar devices were used to track vehicles at intersections. Each radar measured x and y position and speed with a frequency of 4.4 Hz, that is, 0.2275 s time interval. Furthermore, radars automatically assigned IDs to each vehicle for each measurement based on an internal algorithm using past observations. Traffic light data were also available and used in this project. Figure 3 depicts several tracked vehicle trajectories tracked by radars.

The quality of the raw data collected by the radars presented two main challenges:

(i) The raw data was full of gaps and, thus, incomplete. Occlusion of the radars and interference and reflections from the city environment were frequently observed, leading to incomplete and split trajectories. These data quality issues represented a major issue as it was not straightforward to assign preceding vehicles to other vehicles, which becomes essential for any CF model formulation.

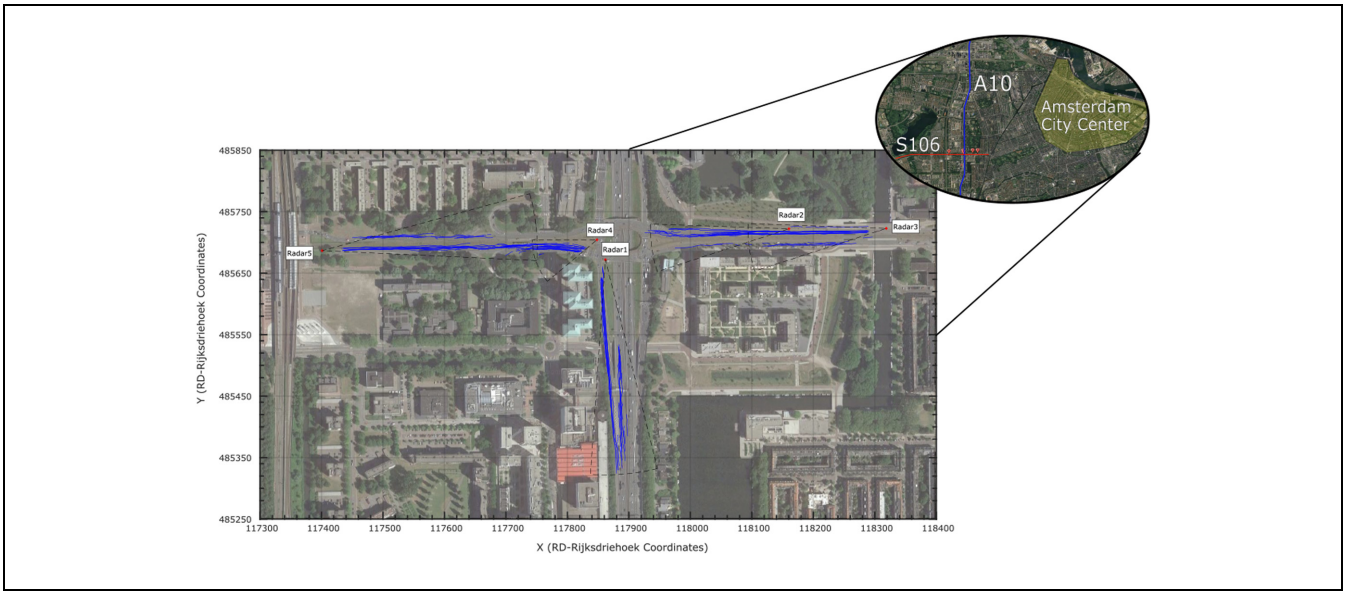

Figure 3. Overview of radars and tracked vehicle trajectories. City background and city map of Amsterdam were retrieved from GoogleMaps (*24*).

(ii) (ii) The data location and speed measurements were noisy because of the radars' nature and also because it was not guaranteed that consecutive measurements of a vehicle were using same vehicle reference position.

To solve all the above-mentioned main issues, several data processes have been carried out; a full description can be found in Echaniz (*25*). In short, trajectories have been independently smoothed per x and y position using the moving average method. Then, split trajectories have been mapped using a linear assignment problem solved by the Hungarian method (*26*) with position, temporal, and lane constraint to assign weights to the trajectory candidates. Later, missing trajectories have been linearly estimated. Those estimations have not been included in the final data set, but they have been used to identify in which order vehicles were driving. If an interpolated (estimation) of a vehicle was either a follower or a leader, the data were discarded. Despite losing 2% of reliable data through wrong estimation, it is avoided that 10% of all reliable points present a wrong preceding assignment.

Finally, from all combinations of reliable consecutive trajectories, that is, leader plus following vehicles, 275 complete coupled trajectories have been randomly selected from the Tuesday 7 and Wednesday 8 June, 2016 from 6 a.m. to 8 p.m. This specific data size is chosen because of computational time constraints of the implementation. Nevertheless, in total more than 32.000 measurements are included into the training set for the GPR, which is considered a big data set.

The quality of the processed data is satisfactory, yet is not ideal for our aim to derive a CF model. First, the vehicle length is unknown. Thus we assume a standard length of 5 m. Second, consecutive radar measurements might not belong to identical vehicle points. For instance, the traffic radar can measure the location of a vehicle taking as a reference the front windows of a vehicle, whereas the consecutive measurement takes the front bumper as a reference. This issue is even worse if the road is not completely perpendicular to the radar, as in this case. These two aspects have a great influence in the calculation of the spacing in between vehicles. By smoothing the trajectories in the data processing the impact of these data peculiarities has been reduced. However, still relatively small spacing (< 0.5 m), and collisions are even observed in real observations. Collisions have been excluded from the data. Nonetheless, the dilemma remains whether small spacing should be deleted or not. In this study, we use all positive spacing measurements as we cannot ensure that small spacing measurements are not reliable. This data aspect has had a great influence on the results, as explained in the simulation results section.

Assessing the Model's Quality

The reliability and robustness of the different GPR variable combinations fits are assessed by applying specific performance criteria, which measure the deviations between the GPR simulated predictions and the real data. To find the robust model, one main key performance indicator (KPI) is used to benchmark the different set of models: the spacing mixed error. This is presented as follows:

$$\mathscr{F}_{\text{mix}}[s^{\text{sim}}] = \sqrt{\frac{1}{\langle |s^{\text{data}}| \rangle} \left\langle \frac{(s^{\text{sim}} - s^{\text{data}})^2}{|s^{\text{data}}|} \right\rangle} \tag{11}$$

As introduced in implementation of the methodology, the mixed error, that is, $\mathscr{F}_{\text{mix}}$, is also used in the optimization scheme. This error measures the percentage difference between consecutive trajectories in relation to spacing (*27*). Each simulated trajectory is initialized by the initial position and speed of the observed trajectory from the validation data. From the second time step, the simulated trajectory becomes independent from the observed trajectory and exclusively depends on itself and the observed leader. Each time step, the difference in spacing between the observed and the simulated trajectory is computed. Later, a temporal average is done according to the amount of the data points of the trajectory. Finally, the average from different mixed error of every specific trajectory is calculated. As shown in the formulation of this KPI, s^{data} is partially excluded from the main term. Kesting and Treiber (*27*) suggest this approach to avoid overestimation errors for large gaps at high velocities, and to avoid systematic overestimates of deviations of the observed headway in the low velocity range. The conceptual interpretation of the mixed error is the percentage error between the simulated and the observed trajectory. Previous studies show that best fitting models present a mixed error in a range of 10%–30% (*27–30*).

The GPR models are not collision free; that is, its mathematical formulation does not explicitly ensure no collisions. Thus, the number of collisions observed in the simulated trajectories is also used as a KPI.

A complete day data set of one lane is exclusively used for validation purposes. Thus, this data set is not included in the training data set. Specifically, data from June 9, 2016 are used. Examples of these trajectories can be seen both in Figures 3 and 4. Dry weather conditions were observed that specific day. In total, 2,790 individual following trajectories with a minimum of 10 continuous measurements are used to assess the models. Note that both consecutive following trajectories and leader trajectories are needed. These trajectories are exclusively used for validation purposes (i.e., not training). The data set

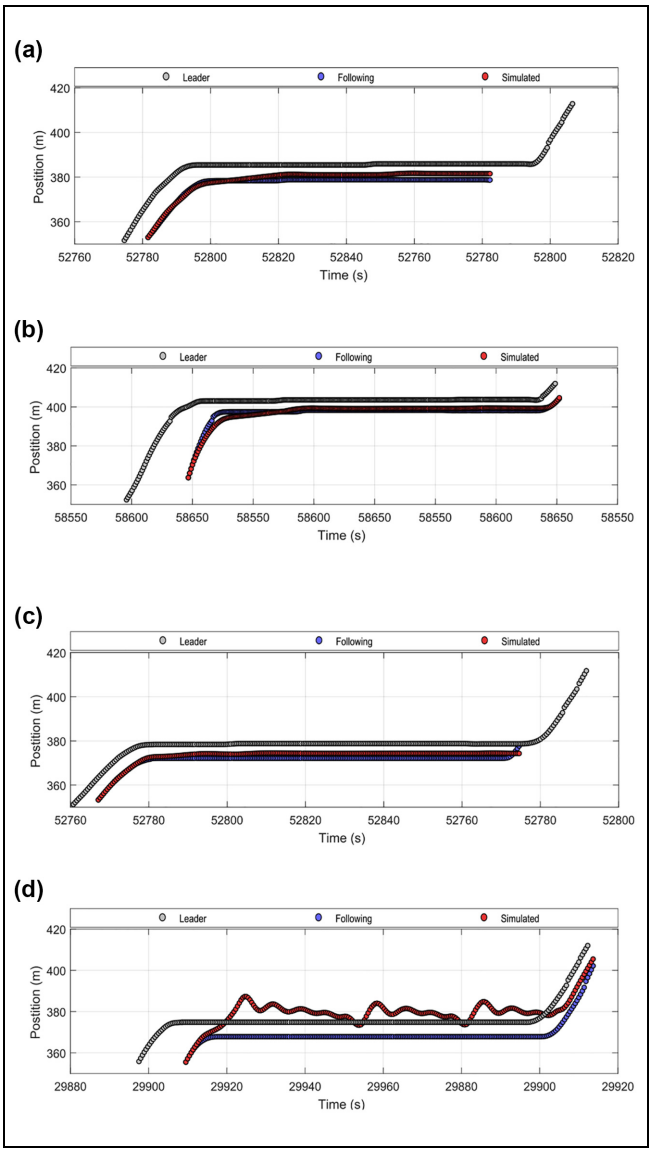

Figure 4. Simulated versus observed trajectories. Gaussian process regression (GPR) model trained with spacing, speed, speed difference and status of the traffic light as predictor variables and a fixed basis optimal velocity model function. (*a*) Example of a good trajectory estimation by the GPR (deceleration). (*b*) Example of a good trajectory estimation by the GPR (deceleration + acceleration). (*c*) Example of a good trajectory estimation by the GPR (deceleration). (*d*) Example of a wrong trajectory estimation by the GPR leading to a collision.

size for training purposes is smaller than the data set used to assess the model's quality (279 versus 2,790 trajectories). The main reason for this is computation time. A larger training set would be infeasible to compute. However, note that in the 279 trajectories in the training set, there are more than 32,000 observation points in total. We could have used a smaller validation data set following traditional division of data sets (e.g., 80/20); however, we believe that testing for a larger data set can lead to a more reliable validation process, particularly to assess the collisions. By performing an empirical validation with the before mentioned KPI, we are exclusively assessing the validity of the model inside the data space region of the validation (and training) data, for example, a road section of 100 m in an urban signalized intersection.

Case Study Set-up

We have trained several GPR models using different predictor variables combinations. These combinations of variables can be found in Table 1. As a constraint for using a basis function, spacing $s_{(n,\,n-1)_{t-1}}$ and speed $v_{n_{t-1}}$ need to be always included in all combinations. After training all GPR models with the training data, we have simulated a full day of trajectories and validated each GPR model individually, as depicted in the previous subsection. By doing so, we can assess the validity of each model and also the influence of each predictor variables on the driver behavior. As is depicted in Algorithm 1, we assume the predictive mean acceleration of the GPR to predict the vehicle's trajectories. Finally, we have compared the best GPR fitted model with a calibrated OVM using the same training set. Therefore, we check whether adding non-parametric formulation to an existing parametric model improves the accuracy of the predictions. Unfortunately, we are not able to compare the GPR model with other data-driven techniques, as it not feasible because of the lack of formulation studies available and its complexity. It is left for future research.

Results

GPR Models and Variables Relationships

Table 1 depicts a set of GPR models trained combining different predictor variables: spacing $s_{(n,\,n-1)_{t-1}}$, speed $v_{n_{t-1}}$, speed difference $\Delta v_{(n,\,n-1)_{t-1}}$, the distance to the traffic light $x_{n_{t-1}}$, and status of the traffic light $STAT_{t-1}$. In total, eight models are tested and evaluated in an arbitrary order according to the KPIs and validation data described in previous section. Generally speaking, the results show that several GPR models are able to make accurate predictions, that is, $\mathscr{F}_{\text{mix}}$ smaller than 20%. However, all GPR models present collisions. For example, there are nearly 200 collisions in the best models. This represents a collision in 6% of the trajectories in the validation set. Spacing and speed are two major variables in all CF models. We cannot study their influence independently because of the constraints of the basis function. The GPR model with both variables has a mix error of 30%, showing that there is the need to include more explanatory variables. Adding speed difference seems essential to describe driver behavior at urban

Table 1. Gaussian Process Regression Models' Results Trained Combining Different Predictor Variables

		Variables					Results	
							Fixed basis function	
		s	v	Δv	x	STAT	$\mathscr{F}_{\text{mix}}[spacing]$ (%)	Collisions *
1	2 Variables	✓	✓	-	-	-	38	811
2	3 Variables	✓	✓	✓	-	-	19	177
3	3 Variables	✓	✓	-	✓	-	35	774
4	3 Variables	✓	✓	-	-	✓	33	659
5	4 Variables	✓	✓	✓	✓	-	21	634
6	4 Variables	✓	✓	✓	-	✓	17	256
7	4 Variables	✓	✓	-	✓	✓	31	267
8	5 Variables	✓	✓	✓	✓	✓	18	192

NOTE: ✓ = selected; - = Not selected. *Collision out of the total number of following trajectories in the data set (2,790).

signalized intersections. When this variable is added, both the number of collisions and the $\mathscr{F}_{\text{mix}}$ are drastically reduced. Similarly, adding traffic light status improves the model accuracy; that is, lower $\mathscr{F}_{\text{mix}}$ and fewer collisions. Finally, distance to the traffic light seems to not influence the acceleration of drivers. This variable even worsens the results. Note that these results only refer to the space region of the data set. Therefore, assessment is performed according to how similar the model is compared with the validation data.

Even in the best GPR trained models, simulation shows that collisions still occur (6% of the simulated trajectories). Figure 4 shows several simulated trajectories. The GPR model is able to predict accurate deceleration of drivers until standstill, as depicted in sub-figures 4, *a* or *b*. Moreover, when the traffic light turns green and the leader vehicle starts accelerating, the GPR model predicts positive acceleration values (e.g., sub-figure 4*b*). Collisions are mainly predicted by the GPR model when following drivers are driving relatively close to the leader (small spacing values), such as in sub-figure 4*c*. When the leader stops at a traffic light, the model is sometimes not able to predict high deceleration values. This leads to situations in which vehicles are not able to brake in time. The main hypothesis to explain the collisions is that spacing measurements are sometimes not reliable because of the noisy position measurements collected by the radar. In the training set, very small spacing measurements (i.e., smaller than 0.5 m) with a great variability of speeds and speed difference values can sometimes be observed, which seems unrealistic. All these lead to high variance and an inaccurate predicted mean acceleration when spacings are relatively small ($\leqslant 2$ m), which are usually found in deceleration phases. In this study we assumed that each measurement belonged to the front part of a vehicle. Furthermore, we assumed a length of 5 m for all vehicles. To obtain the net spacing, the length of a vehicle was subtracted from the distance measured between consecutive drivers' position measurements. However, in reality, the radar might be measuring other parts of the vehicle, such as the side part of a vehicle instead of the front point. Also, the vehicle length in The Netherlands varies significantly among vehicles. Typically this varies from 3.5 to 5.4 m for passenger cars and vans. Trucks and semi-trucks measure up to 12 m and 18.75 m, respectively. Moreover, by smoothing the position measurements in the data-processing section we are altering the reference point. With all these issues, we might be committing an error of dozens of centimeters or even few meters. Therefore, if the predicted trajectories are also off by a couple of dozen centimeters, this will lead (directly or indirectly) to predicted collisions. Therefore, reliable spacing measurements are essential for any CF to simulate stop-and-go traffic conditions and to avoid collisions. Having more reliable training data would presumably lead to models with no collisions. To avoid model collisions we could include an extra parametric formulation to enforce a collision-free model. Another approach to improve the performance of the model could be using clustering methods as a basis for a (meta-)model, whose variables are not either of the base variables but some combined ones.

Performance of GPR Formulation versus Parametric Formulation Stand-Alone

This section compares the performance of GBR formulation versus a parametric model stand-alone. This is done to see if there is a benefit of using GBR formulation compared with traditional models and techniques. We have used the OVM as a parametric model. In other words, we are actually checking whether adding non-parametric formulation to an existing parametric model improves the accuracy of the acceleration predictions. The OVM has been separately calibrated using a

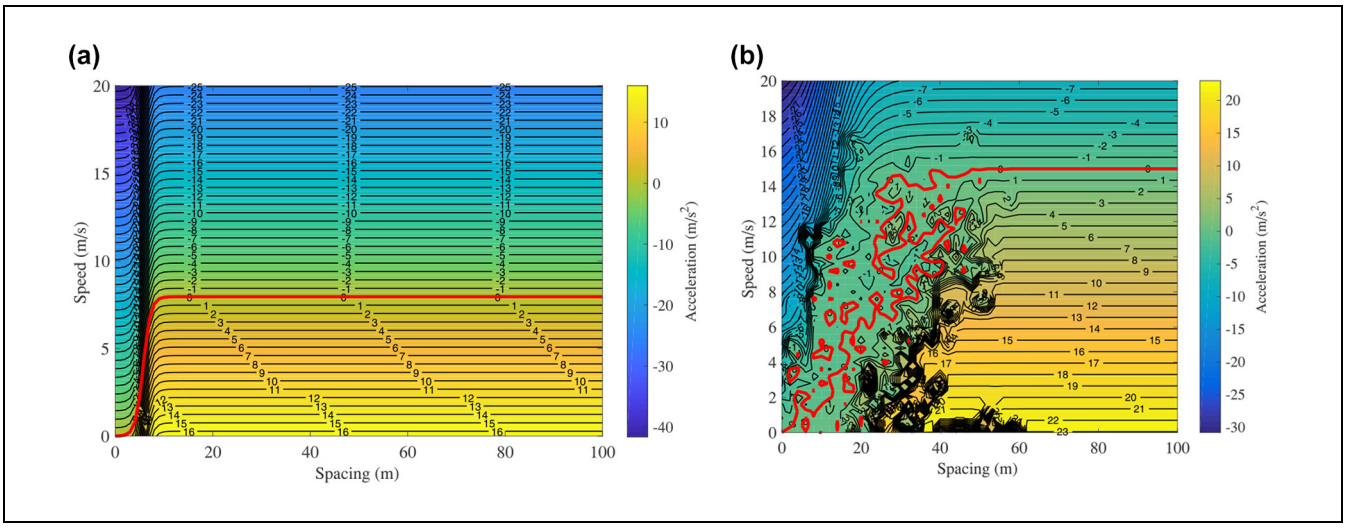

Figure 5. Gaussian process regression (GPR) results. Predicted acceleration of calibrated optimal velocity model (OVM) and GPR results using speed and speed difference. Red line shows the speed–spacing combinations where acceleration is zero. (*a*) Calibrated OVM; (*b*) GPR results.

traditional nonlinear optimization problem and a classic optimization algorithm, that is, interior point. We have used the same training of the GPR models to calibrate the OVM stand-alone. Initial parameters and parameters constraints in the optimization were set to the values according to Treiber et al. (*31*).

Results show that OVM stand-alone scores worse than the best GPR model in relation to the mixed error, that is, 17.4 % versus 18.6%. Nonetheless, OVM stand-alone presents no collisions, whereas the best GPR model presents around 175 collisions out of 2,790 trajectories. Figure 5*a* shows the acceleration predicted by the OVM given spacing and speed values and using the optimal parameters found in the calibration. According to the results, the calibrated OVM is quite accurate in similar space regions of the data set and presents typical mixed error ranges found in the literature (*27*). Yet, outside space regions of the data set, the acceleration predictions are poor. The model seems to suffer of overfitting in the space regions of the calibration data. According to the results of the OVM stand-alone, if spacing is big enough, drivers accelerate until achieving a desired speed of 28 km/h. Afterward, drivers keep their speed constant (zero acceleration). A desired speed of 28 km/h represents a low value to describe the desired speed in a signalized intersection with a speed limit of 50 km/h. Therefore, it represents a wrong model estimation in relation to completeness.

Figure 5 exemplifies both options: traditional calibration of the OVM (basis function) stand-alone versus GPR formulation. The figures show the acceleration predicted by the calibrated OVM and by the GPR built with the same explanatory variables (speed and spacing). Figure 5*a* shows the simple prediction shape of the OVM, which results in a relatively low desired speed. Figure 5*b* clearly depicts the space regions where there is historical data (complex shape) and the regions where the model relies in the OVM (similar to left figure). To highlight, the transition between the parametric and non-parametric model plays a major role in this kind of GPR model. Note that the GPR model trained with only speed and spacing as in this figure presents a poor mixed error—38%—as shown in Table 1. The OVM stand-alone presents a mixed error of 18.6%. Higher accuracy models, which score better than the OVM stand-alone, are achieved with four predictive variables, which would be challenging to represent in a graph. Overall, this example shows the importance of the GPR formulation to ensure a complete model and avoid overfitting issues of the traditional parametric calibration techniques.

Conclusions

This paper has proposed a methodology to derive a CF model based on GPR formulation with a basis function. The main scientific contribution of this paper is to explore the application of the GPR formulation to describe longitudinal driver behavior to benefit from large data sets that are yet incomplete and from small space regions. The main weakness of data-driven models to describe driving behavior is that data for all driving conditions in all driving conditions are necessary, which are often lacking for rare situations such as collisions or strange driver behavior. This gap is filled by combining

traditional parametric models and non-parametric in the GPR formulation.

This paper shows that this methodology is able to provide accurate acceleration predictions. By using GPR formulation we obtained new variables relationships to describe longitudinal behaviors of cars. This represents a major benefit, particularly for formulation of CF models, as traditionally traffic engineers have not extensively explored variables such as the traffic light or the distance to the traffic light. In this paper, we show that given the data set we possess (Amsterdam NL), the status of the traffic light influences the acceleration of drivers. This would have been difficult to check with other formulations and techniques. Next to the status of the traffic light, spacing, speed, and speed difference with the leader are the other explanatory variables included in the most accurate GPR model.

Despite results showing that GPR models are as good as traditional CF models (*27*), collisions are still occurring. Therefore, it is expected that with full non-parametric approach, rare events would occur even more often because of extreme and wrong acceleration prediction. This highlights the need to design hybrid models with good transitions to parametric models to guarantee that certain rare and unexpected events do not occur. In any case, if new data sets with new predictor variables or from other space regions become available, the model can be easily upgraded.

We have analyzed if there is an improvement by applying GPR formulation to turn a parametric model into a hybrid parametric and non-parametric model. We have calibrated the optimal velocity CF model (OVM) used in the basis function of the GPR with the same training data used to trained the GPR models. A GPR model with fixed basis function scores better results than OVM in relation to mixed error. However, still the original OVM ensures no collisions, whereas the GPR model occasionally predicts collisions. The GPR solution is the predictive mean and the variance of the acceleration. Trajectories are estimated from the predictive mean only. How the variance can be included in the estimation of trajectories and how to improve the accuracy of the acceleration predictions is an interesting line of research for future studies. The solution is directly derived from the hyper-parameters, which includes the noise of the measurements. The noise and measurement errors from the spacing might be too significant to ensure a collision-free model, especially given that spacing is one of the most important variables in CF formulation to replicate stop-and-go traffic conditions.

This modeling methodology allows us to build accurate models in space regions where we possess data and, at the same time, ensure completeness in all space regions. This study shows a methodology to describe driving behavior which can be updated over time without new enumerations of the model as soon as new data become available. This proves how strong this formulation can be for the transportation field.

Acknowledgments

The authors gratefully acknowledge FILERADAR (The Netherlands) for providing the large data set to perform this project.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: I. E. Soldevila, V. L.Knoop, and S. Hoogendoorn; analysis and interpretation of results: I. E. Soldevila, and V. L.Knoop; draft manuscript preparation: I. E. Soldevila, and V. L.Knoop. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Ignasi Echaniz Soldevila https://orcid.org/0000-0002-9553-638X
Victor L. Knoop https://orcid.org/0000-0001-7423-3841
Serge Hoogendoorn https://orcid.org/0000-0002-1579-1939

References

1. Olstam, J. J., and A. Tapani. *Comparison of Car-following models*. Report VTI Report 960A. Swedish National Road and Transport Research Institute (VTI), 2004.
2. Brackstone, M., and M. McDonald. Car-Following: A Historical Review. *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 2, No. 4, 1999, pp. 181–196.
3. Treiber, M., and A. Kesting. Traffic Flow Dynamics. *Traffic Flow Dynamics: Data, Models and Simulation*. Springer-Verlag, Berlin, Heidelberg.
4. Federal Highway Administration US. *FNext Generation Simulation (NGSIM)*, 2017. https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.
5. Daamen, W., C. Buisson, and S. P. Hoogendoorn. *Traffic Simulation and Data: Validation Methods and Applications*. CRC Press, Boca Raton, FL, 2014.
6. Ossen, S. J. L. *Longitudinal Driving Behavior: Theory and Empirics*, Vol. 2008. Netherlands TRAIL Research School, 2008.
7. Karlaftis, M. G., and E. I. Vlahogianni. Statistical Methods Versus Neural Networks in Transportation Research:

Differences, Similarities and Some Insights. *Transportation Research Part C: Emerging Technologies*, Vol. 19, No. 3, 2011, pp. 387–399.
8. Hofleitner, A., R. Herring, and A. Bayen. Arterial Travel Time Forecast with Streaming Data: A Hybrid Approach of Flow Modeling and Machine Learning. *Transportation Research Part B: Methodological*, Vol. 46, No. 9, 2012, pp. 1097–1122.
9. Elfar, A., A. Talebpour, and H. S. Mahmassani. Machine Learning Approach to Short-Term Traffic Congestion Prediction in a Connected Environment. *Transportation Research Record: Journal of the Transportation Research Board*, 2018. 2672(45): 185–195.
10. Lv, Y., Y. Duan, W. Kang, Z. Li, and F.-Y. Wang. Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 2, 2015, pp. 865–873.
11. Panwai, S. and H. Dia. A Reactive Agent-Based Neural Network Car Following Model. In *Proc., 2005 IEEE Intelligent Transportation Systems*, IEEE, 2005, pp. 375–380.
12. Khodayari, A., A. Ghaffari, R. Kazemi, and R. Braunstingl. A Modified Car-Following Model Based on a Neural Network Model of the Human Driver Effects. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, Vol. 42, No. 6, 2012, pp. 1440–1449.
13. Wang, X., R. Jiang, L. Li, Y. Lin, X. Zheng, and F.-Y. Wang. Capturing Car-Following Behaviors by Deep Learning. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 3, 2017, pp. 910–920.
14. Zhu, M., X. Wang, and Y. Wang. Human-Like Autonomous Car-Following Model with Deep Reinforcement Learning. *Transportation Research Part C: Emerging Technologies*, Vol. 97, 2018, pp. 348–368.
15. Gao, H., G. Shi, G. Xie, and B. Cheng. Car-Following Method Based on Inverse Reinforcement Learning for Autonomous Vehicle Decision-Making. *International Journal of Advanced Robotic Systems*, Vol. 15, No. 6, 2018, p. 1729881418817162.
16. Lee, S., D. Ngoduy, and M. Keyvan-Ekbatani. Integrated Deep Learning and Stochastic Car-Following Model for Traffic Dynamics on Multi-Lane Freeways. *Transportation Research Part C: Emerging Technologies*, Vol. 106, 2019, pp. 360–377.
17. Yang, D., L. Zhu, Y. Liu, D. Wu, and B. Ran. A Novel Car-Following Control Model Combining Machine Learning and Kinematics Models for Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 6, 2018, pp. 1991–2000.
18. Papoulis, A., and S. U. Pillai. *Probability, Random Variables, and Stochastic Processes*. Tata McGraw-Hill Education, 2002.
19. Cressie, N. The Origins of Kriging. *Mathematical Geology*, Vol. 22, No. 3, 1990, pp. 239–252.
20. Snelson, E. L. *Flexible and Efficient Gaussian Process Models for Machine Learning*. University of London, University College London, 2008.
21. Rasmussen, C. E., and C. K. Williams. *Gaussian Processes for Machine Learning*, Vol. 1. MIT Press Cambridge, 2006.
22. Bando, M., K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama. Dynamical Model of Traffic Congestion and Numerical Simulation. *Physical Review E*, Vol. 51, No. 2, 1995, p. 1035.
23. Hoogendoorn, S., R. Landman, J. Van Kooten, and M. Schreuder. Integrated Network Management Amsterdam: Control Approach and Test Results. *Proc., 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, The Netherlands, 2013, IEEE, pp. 474–479.
24. GoogleMaps. Google Maps: City of Amsterdam. https://www.google.com/maps/@52.373511, 4.865366,13z, 2018.
25. Echaniz, I. *Car-Following Model using Machine Learning Techniques: Approach at Urban Signalized Intersections with Traffic Radar Detection*. Master's thesis. Delft University of Technology, 2017.
26. Kuhn, H. W. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, Vol. 2, No. 1–2, 1955, pp. 83–97.
27. Kesting, A. and M. Treiber. Calibrating car-following models by using trajectory data: Methodological study. *Transportation Research Record: Journal of the Transportation Research Board*, 2008, 2088: 148–156.
28. Brockfeld, E., R. D. Kühne, and P. Wagner. Calibration and Validation of Microscopic Traffic Flow Models. *Transportation Research Record: Journal of the Transportation Research Board*, 2004. 1876(1): 62–70.
29. Ranjitkar, P., T. Nakatsuji, and M. Asano. Performance Evaluation of Microscopic Traffic Flow Models with Test Track Data. *Transportation Research Record: Journal of the Transportation Research Board*, 2004. 1876(1): 90–100.
30. Punzo, V., and F. Simonelli. Analysis and Comparison of Microscopic Traffic Flow Models with Real Traffic Microscopic Data. *Transportation Research Record: Journal of the Transportation Research Board*, 2005. 1934(1): 53–63.
31. Treiber, M., A. Hennecke, and D. Helbing. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E*, Vol. 62, No. 2, 2000, p. 1805.