Stochastic Range Estimation Algorithms for Electric Vehicles using Data-Driven Learning Models

Zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN
(Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)
genehmigte

DISSERTATION

von

M.Sc. Stefan Scheubner
geb. Sautermeister

Tag der mündlichen Prüfung:	23. April 2021
Referent:	Prof. Dr. rer. nat. Frank Gauterin
Korreferent:	Prof. Dr.-Ing. Peter Vortisch

Abstract

Limited range and charging infrastructure leads to range anxiety of electric vehicle drivers. Current range estimation algorithms are deemed unreliable and large safety margins are reserved to prevent the risk of stranding. This is one of the main problems limiting widespread acceptance of electric vehicles. One approach to reduce range anxiety is a more reliable range estimation, which in general depends on two factors: current battery energy content and the energy consumption forecast on the route to destination. This work aims at improving the latter by enhancing the forecast with a notion of uncertainty. The prediction algorithm itself learns from driver and traffic data in a training set to generate accurate, driver-individual energy consumption forecasts. Thereby, a central part of the algorithm is the explicit evaluation of the traffic situation by classifying the traffic phases. With the help of this methodology, individual forecasts can be made more precise since they are highly dependent on the surrounding traffic. To demonstrate the validity of the algorithms, the performance is evaluated using real test drive data comprising multiple drivers. On the basis of the performance evaluation, both the superiority of stochastic algorithms over deterministic predictions and the improvement of predictive performance by evaluating explicit traffic phases can be shown. Implementing the proposed methodology in modern day electric vehicles could reduce range anxiety and ultimately increase acceptance of electric mobility worldwide.

Kurzfassung

Stochastische Algorithmen zur Berechnung der Reichweite von Elektrofahrzeugen mittels datengetriebener Lernmodelle
Eingeschränkte elektrische Reichweite und Ladeinfrastruktur führen zu Reichweitenangst bei Fahrern von Elektrofahrzeugen. Aktuelle Algorithmen zur Berechnung der Reichweite werden als unzuverlässig empfunden und hohe Sicherheitspuffer werden reserviert, um das Risiko eines Liegenbleibers zu reduzieren. Dies ist eines der Hauptprobleme, welches eine weitreichende Akzeptanz von Elektrofahrzeugen verhindert. Ein Ansatz um die Reichweitenangst zu verringern ist eine zuverlässigere Reichweitenberechnung, welche grundsätzlich auf zwei Faktoren beruht: dem aktuellen Energieinhalt der Batterie und der Energieverbrauchs-Vorhersage auf der Route bis zum Ziel. Diese Arbeit zielt darauf ab, letzteres durch das Berücksichtigen von Vorhersage-Unsicherheiten zu verbessern. Der Prädiktionsalgorithmus selbst lernt von Fahrer- und Verkehrsdaten aus einem Trainings-Set um akkurate, fahrerindividuelle Energieverbrauchsvorhersagen zu generieren. Dabei ist die explizite Berücksichtigung der Verkehrssituation durch Klassifikation der Verkehrsphasen ein zentraler Aspekt des Algorithmus. Mithilfe dieser Methode können individuelle Vorhersagen präziser berechnet werden, da sie abhängig vom Verkehr in der Umgebung sind. Um die Algorithmen zu validieren wird deren Performance mittels Realfahrdaten von verschiedenen Fahrern evaluiert. Auf Basis dieser Evaluation kann sowohl die Überlegenheit von stochastischen Algorithmen gegenüber deterministischen Vorhersagen als auch die Verbesserung der Vorhersagegüte durch Berechnung expliziter Verkehrsphasen gezeigt werden.

Acknowledgements

This dissertation was created during my time at the research facility in Weissach of Dr. Ing. h.c. F. Porsche AG. A cooperation between Porsche's department of energy management and the KIT Institute of Vehicle System Technology made this project possible.

First of all, I would like to sincerely thank *Professor Frank Gauterin*, who was my adviser offering his support and scientific guidance regarding the dissertation. He gave me the right methodical impulses and showed me whether I was on the right track by his feedback.

Furthermore I like to thank *Professor Peter Vortisch* for being my second adviser and *Professor Barbara Deml* for chairing the defense committee.

I am enormously grateful to *Moritz Vaillant*, who was my adviser from Porsche. Listing all the things he has done for me would not be possible. He gave me scientific input, helped me acquiring necessary resources, offered personal support and was always available in case of a problem.

Special thanks also goes to *Michael Frey* who took a lot of time for methodical discussions in Karlsruhe.

I also thank *Marc Albrecht* for the research opportunity in his department at Porsche. He was a trusting and supportive supervisor, who was always there for me when I needed his help. Hiring me as a development engineer after the PhD program gave me the chance to learn the reality of series-implementation and leave my fingerprints on Porsche production vehicles instead of just prototypes.

The test drives in this work would not have been possible without *Volker Watteroth* and his practical genius. I came to him with many problems I

encountered concerning the prototypes' control boxes, but Volker solved them all. He was always helpful and open for my issues. I dare say some test drives with him also were a lot of fun...
I was blessed with many exceptional colleagues at Porsche, both among other PhD candidates and department members. *Sebastian Fünfgeld* was both and I cannot thank him enough for the impulses regarding my own research as much as his mathematical genius and patient explanations. With Sebastian, many coffee breaks were intense scientific discussions that brought my work big steps forward.
I would like to thank all the students who wrote a Bachelor or Master Thesis within the project: *Felipe Peter*, *Lucas Maser*, *Philipp Bock*, *Florian Ott*, *Adam Thorgeirsson* and *Bruno Closs*. Special thanks goes to Adam and Florian who became my co-authors thus helping me get published.
Max Falk was another one helping me get published by being my co-author on the first paper I wrote. Max taught me the magic of LateX for which I am very grateful.

From the beginning of my education I could count on the support of my parents. In decisive moments of my life, they showed me the right choices for which I am deeply grateful.

Lastly I thank my wife *Mirjam Scheubner* who always lifted me up in frustrating situations, helped me relax in stressful times and suffered my growing love for statistics in everyday life.

Karlsruhe, 23.04.2021 *Stefan Scheubner*

Contents

1 Introduction

1.1 Motivation

In recent years, research interest in electric mobility inceased because it offers the chance to reduce emissions worldwide [75, 19], especially local emissions in urban areas [120] and in combination with renewable energy sources [114]. Thereby, emission reduction potential is primarily focused on gaseous emissions such as NO_x or CO_2, the latter is of special importance in the background of global climate change. A further aspect is the noise reduction potential of electric vehicles in cities [76]. In consequence, policy makers support electric mobility, e.g. by creating incentives for buyers, and further tighten emission regulations for passenger vehicles worldwide. Some governments and cities have passed legislation for a complete ban of internal combustion engine vehicles (ICEVs) [25] and nowadays it seems that until wide-range usage of pure-electric vehicles is merely a matter of time. In fact, vehicle manufacturers have started to bring battery-electric vehicles (BEVs) to the mass market, their sales numbers having increased exponentially in recent years [54].

However, the energy density of the battery in an electric vehicle is significantly lower than in fossil fuels [85]. This disadvantage cannot be balanced by the superior efficiency of an electric powertrain compared to an ICEV leading to limited driving range of BEVs. In addition, with the current charging infrastructure, BEV charging time is much higher than it takes to refuel an ICEV and on top of that, there are fewer electric charging stations than gas stations. Consequently, drivers of electric vehicles experience *range anxiety*, which is the fear of stranding with an empty battery [26]. To reduce this effect and help increase acceptance of electrified vehicles, batteries with growing maximum capacities are installed in the cars, up to 100 kwh nowadays in the TESLA models [124]. Additionally, increased efforts to build a dense, powerful network of charging stations are taken by electricity providers and car manufacturers [97]. Nonetheless, it will take time until driving range and charging infrastructure is sufficient and BEV drivers do not have to worry about stranding anymore.

Apart from hardware solutions, intelligent driver assistance systems such as *eco-routing* [83] and *charge-planning* [72, 18, 125, 126] are developed to reduce range anxiety. Thereby, eco-routing helps the driver to consume less energy and thus increase range whereas charge-planning selects charging stations along the planned route to allow long-distance trips with a BEV. Those software solutions are based on a range estimation algorithm which calculates the answer to the core question: *How far can the vehicle drive until the battery runs out of energy?* If the range estimate is not reliable, eco-routing or charge-planning strategies might fail or cause unnecessary long trips. The presented dissertation in the following focuses on algorithms and models for reliable range estimation to help reduce range anxiety and thus increase acceptance of electric vehicles.

1.2 Problem Statement and Research Questions

Range estimation in electric vehicles provides information for the range display as well as the advanced driver assistance systems mentioned in the previous section. It can be imagined that if it is unreliable, driver's would not trust the range display anymore and eco-routing or charge-planning strategies might fail or cause unnecessary long trips. Thus, improving the accuracy of range estimation algorithms is the focus of a lot of research activity in the respective field and an overview is given in Section 3.1. Traditionally, the output is a deterministic quantity signifying range as *distance-to-empty*.

Prediction accuracy is consequently measured based on the difference between predicted range and the distance driven until the battery is empty. However, range estimation includes a forecast into the future of the trip, because the future energy consumption must be obtained.

Thereby, any forecast in a real-world problem is subject to uncertainty, especially when involving human behaviour. In consequence, deterministic predictions will rarely be exactly equivalent to the true value, instead they are scattered around it according to the unknown uncertainty. In this work, an important differentiation is made between *accuracy* and *reliability*: accuracy means how close the prediction is to the measurement and reliability means whether the accuracy information is included in the forecast. Thus, the reliability definition is derived from the statistical interpretation of consistency. By this definition, deterministic forecasts can be accurate but not reliable which is one of the reasons why range displays in modern electric vehicles are deemed unreliable [33]. This problem can be avoided by calculating system uncertainties and including them in the range estimate as a stochastic forecast. In fact, statisticians urge the necessity for stochastic forecasts for real-world applications e.g. in [39].

Another problem for range estimation is the variety of influence factors, ranging from environmental influences (weather, road conditions) over vehicle-specific influences (powertrain efficiency, weight) to human influence (driving behaviour, traffic state). Among those, vehicle-specific and environmental influences must be modelled using the known physical dependencies and available vehicle sensors. Thereby, the different characteristics of each specific variable and limited computing power and sensor information pose a challenge for the researcher. However, another challenge is induced by driving behaviour and traffic state because they are non-deterministic influence factors which cannot be explained on a pure physical level. Instead, they must be learned using data and appropriate machine-learning models.

From these challenges, research questions are derived, which serve as guidelines in the course of this dissertation. The first one rises from the problem of reliable forecasts including a measure of uncertainty:

RQ1: *How can electric vehicle range be expressed stochastically to signify the amount of uncertainty in the system?*

Thereby, **RQ1** aims at finding a high level approach high level approach for an adequate mathematical expression first before developing a specific algorithm. Apart from the overall methodology, the multitude of influence factors has to be included in the forecast model. While many of the physical models already exist in literature and have to be applied accordingly, the non-deterministic problem of driver and traffic needs to be solved as well. An aspect only partly covered in research is the inter-dependency of driver and traffic which creates a system in which some states are only partly observable. For example, driving behaviour is not dominant in a dense traffic jam and in turn, the speed of surrounding vehicles is of secondary importance in a free-flow scenario. Therein lies the second research question:

RQ2: *How can driving behaviour depending on the traffic situation be determined in the ego-vehicle to include it explicitly in the range estimation forecast?*

Thereby, the *ego-vehicle* means the individual vehicle for which the range estimation is made. Although **RQ2** is focused on the problem of driver and traffic, the second part includes the physical influences as well since without them, driving range cannot be calculated. It can be imagined that a multitude of models and model-combinations can be developed to solve **RQ2**. Hence, the natural question arises of which model-structure is the best:

RQ3: *Which of the considered model-structures yields the best overall performance in a range estimation context?*

To answer these questions, especially **RQ3**, an appropriate test dataset must be obtained. In this dissertation, the performance evaluations are all based on real test drives with modern production cars and different drivers to capture real-world conditions. The question of whether the test drive data is enough for statistical significance, basically *how much testing is enough?* lies outside of this work's focus.

1.3 Structure of this Dissertation

Dissertations often include an extensive chapter introducing the required technical background. This dissertation however illustrates only the main basics in a secluded chapter. As this work covers aspects from many different fields in engineering, the necessary fundamentals are explained in the beginning of the specific sections to improve readability.

After Chapter 1 presented the considered problem of this work as well as the research questions, Chapter 2 illustrates the technical background. That includes energy consumption and driving resistance of an electric vehicle, the characteristics of modern traffic databases, and an overview of forecasting methodologies as well as their evaluation.

Chapter 3 shows the overall approach for range estimation. It starts with an overview of the related research and subsequently, the range estimation approach followed in this work is presented. This approach includes a simplified battery model, which is described in the following before introducing the test drive data required to develop the models. After that, the required post-processing of the data is shown.

In Chapter 4, the first developed algorithms are presented. It starts with the proposed energy consumption model and explains its inherent parameter estimation method. In the second part of this chapter, the algorithm for traffic classification is developed, which provides information to learn driving behaviour. The used approaches are presented and implemented before their performance is compared in the end of the section.

Chapter 5 begins by illustrating the forecast methodology, where the energy consumption model as well as the traffic phase classification provide information to the learning algorithm. The state-of-the-art prediction model is shown, which relies on traffic information. Subsequently, the regression al-

gorithm is illustrated, which poses an alternative model that takes driver behaviour into account. The regression algorithm provides a stochastic velocity forecast. The calculation from velocity to energy consumption requires uncertainty propagation methods, which are presented thereafter. How the stochastic range estimate can be obtained concludes the chapter.
The results of this work are presented in Chapter 6. The proposed approach for performance evaluation is introduced before comparing the algorithms. In the second part of the chapter, the illustrated results are discussed and conclusions are drawn. In Chapter 7, the final remarks of the dissertation are provided, including a summary, the contributions and an outlook.

1.4 Notational and Stylistic Aspects

Although general mathematical conventions are used in this dissertation, some notational aspects shall be introduced in this section. A central aspect of this work is the forecasting methodology. While some researchers use the expressions *forecast* and *prediction* with different meanings, e.g. to imply deterministic or stochastic nature, in this dissertation they are used interchangeably for improved readability. Mathematical symbols are mainly used according to existing literature, therefore bold letters represent vectors, e.g. $\mathbf{x}$, whereas x would be a scalar. A tilde on top of a value denotes a forecast $\tilde{x}$, where deterministic forecasts are indicated with $\tilde{x}^{\mathrm{d}}$ and stochastic forecasts with $\tilde{x}^{\mathrm{s}}$. For models using a specific forecast result, this is denoted in brackets, e.g. $\tilde{y}(\tilde{x}^{\mathrm{s}})$. Furthermore, a bar shows an averaged value $\bar{x}$. If a variable is the result of an estimation process, it is signified with a hat $\hat{x}$.

2 Technical Background

In this chapter, the fundamental technical background necessary for electric vehicle range estimation is presented. The underlying physical theories for electric vehicle energy consumption, namely driving resistances and powertrain losses, are introduced in Section 2.1. The main problem in estimating range addressed in this dissertation is calculating future energy consumption for a specific route, therefore predictive information is required. This can be obtained from the so called traffic and routing database (TRDB), which is explained in Section 2.2. Using that information, a forecast can be made using the methods shown in Section 2.3. Lastly, to assess the forecast performance, validation and evaluation methods are required as illustrated in Section 2.4.

2.1 Energy Demand of an Electric Vehicle

The energy demand of an electric vehicle can fundamentally be divided into two categories: tractive energy E_c and energy consumption of the auxiliaries E_aux. Tractive energy is derived from the power necessary to move the vehicle and depends on the driving resistances as well as the powertrain losses. The auxiliaries are secondary consumers installed in modern day vehicles, e.g. the air-conditioning, seat-heating, active suspension etc. In this dissertation, the focus lies on tractive energy since it represents the main part of total energy consumption [28] and therefore it is important to forecast it accurately. For details on electric vehicle's auxiliaries, see [8] or [50].

Tractive energy consumption is derived from the tractive force a vehicle needs to overcome the driving resistances F_r. These can be described using the standard longitudinal model [55] which is widely used, e.g. in online force estimation algorithms [98], hybrid electric vehicle (HEV) operation strategies [135] or vehicle simulation [128]. In this work, a simplified model is applied which is described in the following. There, details such as toe-in resistance or suspension resistance are not considered. The total driving resistance force F_r consists of the rolling resistance, aerodynamic resistance, acceleration force and climbing force and is calculated with help of (2.1). Rolling resistance is caused mainly by deformation energy of the tyre while the vehicle is moving. It is governed by the rolling resistance coefficient f_r, which depends on vertical force, tyre pressure, temperature, velocity and road surface [55]. Aerodynamic resistance originates from the friction between vehicle surface and surrounding air while the vehicle is moving forward. For a situation without considering wind speed, it depends on the air density ρ, velocity v, frontal area A and the aerodynamic drag coefficient of the vehicle c_w. The acceleration force is caused by translational acceleration a of the vehicle following $F_{acc} = m \cdot a$. Usually, m comprises both vehicle mass and the reduced mass of rotational parts $I_\mathrm{red}/r_\mathrm{dyn}^2$ with r_dyn being the dynamic wheel radius and I_red the reduced moment of inertia. However, the rotational parts are considered in the powertrain model described later in this section. The climbing force originates from the potential energy difference when moving up- or downhill and is defined by the road gradient angle α. More details about the individual resistance forces can be found in [44] or [113]. In (2.1), the driving resistance parameters are concentrated in the vector $\mathbf{p} \in \mathbb{R}^{3\times1}$ whereas the dynamic measurements are included in $\mathbf{B} \in \mathbb{R}^{3\times1}$.

$$F_\mathrm{r} = \underbrace{\begin{pmatrix} m f_\mathrm{r} \\ 1/2\, c_\mathrm{w} \rho A \\ m \end{pmatrix}^{\top}}_{\mathbf{p}} \cdot \underbrace{\begin{pmatrix} g \cos\alpha \\ v^2 \\ a + g \sin\alpha \end{pmatrix}}_{\mathbf{B}} \tag{2.1}$$

Not only the driving resistances F_r determine energy consumption E_c, but also the powertrain efficiency. Efficiency is determined by losses in both mechanical and electrical components in the powertrain. The mechanical losses are caused by friction in the brakes, bearings, drive-shaft, differential and gearbox. The electrical losses arise in the motor and power electronics. For more details about the losses and their causes, see [87]. Commonly, the losses are described with characteristic maps obtained from experiments. The resulting powertrain model depends on the specific vehicle topology and therefore, there is no generic model describing all types of electric vehicles. Figure 2.1 shows one axle of the model for the battery-electric vehicle (BEV) considered in this dissertation to calculate E_c from F_r and v. To give an intuition about the loss magnitude, the percentage of lost energy on a New European Driving Cycle (NEDC) profile is shown. Thereby, losses in the brakes, bearings, drive-shaft and differential are aggregated and indicated with the red arrow at the wheel. Also, power electronics and motor losses are summarized. The resulting energy demand is supplied by the battery, which is presented in Section 3.3. Next to the loss characteristics, the acceleration force required to move the rotational parts is included following [87] and [98]. This exact model was used in an earlier publication [108]. There, the effect of battery losses is also presented, a fact that is not considered in this dissertation.

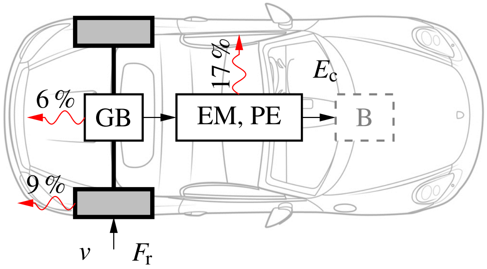


Figure 2.1: Powertrain model with input variables F_r, v and individual component losses of the wheel, gearbox, electric motor and power electronics (percentages exemplary for a NEDC profile). The losses depend on the input variables and ultimately, the total energy consumption E_c can be calculated.

2.2 Traffic and Routing Database

Apart from the energy consumption model explained in the previous section, information about the future trip is necessary to conduct a range estimate. For that, first the destination must be known. If the driver does not enter the destination into the navigation system, it could be predicted based on the previously observed movement data, e.g. [7] or [70]. Subsequently, the range estimate can be obtained for the predicted destination. This dissertation assumes the destination is known by input of the driver, as the energy consumption forecast problem is equivalent for a predicted destination or a chosen one.

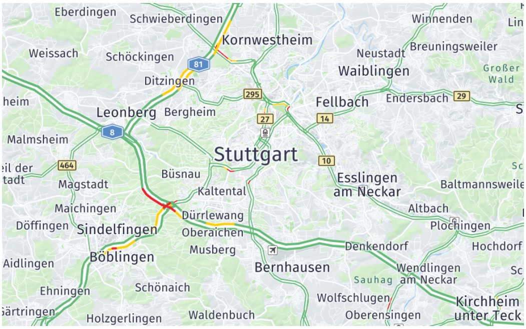
After the destination is selected, the route from the current position to the destination must be obtained. This is done by the conventional navigation systems installed in modern vehicles, typically the user can choose between the *shortest* or *fastest* route, sometimes also an *eco-route*. These options signify the optimization criterion for the routing algorithm, which obtains the result based on map and traffic data [82]. The map divides each street into several segments k (sometimes called links) with a set of individual properties, e.g. length l_k, curvature κ_k, street name, legal speed limit $v_{\mathrm{lim},k}$ etc.

Hence, for the shortest route, the routing algorithm has to find the collection of segments with the lowest total length $\sum_{k=1}^{n} l_k$. The result forms the trip a driver takes to reach the destination.

For the fastest route, the collection of segments with the lowest total duration $\sum_{k=1}^{n} t_k$ must be identified. Thereby, the future travel time $\tilde{t}$ is obtained from the future velocity $\tilde{v}_k$ using fundamental kinematics $\tilde{t} = \frac{l_k}{\tilde{v}_k}$. In consequence, a velocity prediction is required where one option would be to use the legal speed limit, but a more accurate result can be obtained by using traffic data. For that, multiple data sources are used such as loop detectors, cellphones or connected vehicles [142]. All of them provide information about the travel speed of the individual vehicles, called *probes*. A database collecting all probe data then calculates the mean traffic speed $\bar{u}_k$ on each segment k out of the histogram of probe speeds. A known application of those mean speeds is the colouring of congested route segments for cellphone or navigation system users, as shown in Figure 2.2. There, the different levels of congestion are indicated in red, yellow and green depending on the relation of current speed $\bar{u}_k$ to the free-flow speed $\bar{u}_{k,f}$. Free-flow speed is defined as the velocity drivers choose when not being influenced by other vehicles, i.e. when no congestion exists [21]. Based on the mean traffic speed for the whole trip, the routing algorithm can calculate the fastest route. All map and probe information accesible from the car are stored in a traffic and routing database (TRDB) and in this dissertation, the TRDB of the provider HERE is used [48].

2.3 Forecasting Methods

With the energy consumption model from Section 2.1 and the traffic and routing database (TRDB) information presented in Section 2.2, the prerequisites for the range estimation are available. However, to implement a range estimation algorithm, forecasting methods need to be applied because electric vehicle range represents a look into the future of the selected trip. In-



©2018 Here

Figure 2.2: Visualization of the current probe information on major roads around Stuttgart. The level of congestion is indicated in red for jammed segments, yellow for medium congestion and green for free-flow.

tuitively, if a trip takes approximately two hours, energy consumption on segments two hours into the future must be predicted.

Forecasting methods can be divided into two subcategories: *time-based* and *feature-based* [53]. Time-based forecasting uses the characteristics of the variable itself to generate a prediction based on the time-dependent signal of the past. Hence, e.g. velocity v at time $t+1$ depends on v at the previous time-steps:

$$\tilde{v}_{t+1} = f(v_t, v_{t-1}, v_{t-2}, v_{t-3}, \ldots) \,. \tag{2.2}$$

Known examples of this approach are auto-regressive processes, namely AR, ARMA or ARIMA models [13]. For short-term vehicle velocity prediction in the range of a few seconds, such time-based forecast models have been applied, e.g. [57]. Time-series forecasting assumes no relationship between the quantity to predict and other variables. Such a model is advantageous if the system is too complex to model the underlying relationships. See the work of forecasters Hyndman and Athanasopoulos for more information about time-based forecasting [53].

In their book, Hyndman and Athanasopoulos additionally introduce the feature-based forecasting method. They call it *explanatory model* since it is assumed that the quantity to predict has some relationship with other variables, the *features*, which explain the behaviour in the future. Applied on the velocity prediction example used above, velocity v at time $t+1$ would depend on features such as the previous speed v_t, the speed limit v_{lim}, the mean traffic speed $\bar{u}$ and the road curvature κ at $t+1$:

$$v_{t+1} = f(v_t, v_{\text{lim},t+1}, \bar{u}_{t+1}, \kappa_{t+1}, \ldots) \ . \tag{2.3}$$

It can be seen that in this approach, predictive information at $t+1$ about the individual features has to be obtained. In this example, these are static except for $\bar{u}_{t+1}$, therefore a traffic forecast $\tilde{u}$ is required. Feature-based forecasting is widely used, namely in regression or machine learning algorithms such as linear regression or support vector machine (SVM). A common application nowadays is the neural network (NN), which has also been applied to vehicle velocity prediction [91].

Both the time-based and the feature-based forecast can be executed deterministically or stochastically. Deterministic forecasts neglect uncertainty and predict one single value into the future. Stochastic forecasts assess the uncertainty and supplement the forecast with it, thus generating a predictive probability distribution or prediction interval (PI). Paul Saffo argued in one of his publications that uncertainty always has to be assessed for meaningful forecasts:

> The primary goal of forecasting is to identify the full range of possibilities, not a limited set of illusory certainties. [104, p. 1]

Ways to investigate forecast uncertainty can be found in [6], a common approach is treating the forecast quantity as a random variable and use the observed error from past forecasts to express future uncertainty. An example for this method applied on short-term velocity forecast can be found in

[36]. Figure 2.3 illustrates the different forecasts that can be made for $\tilde{v}$. The most common prediction, a deterministic forecast $\tilde{v}^{\mathrm{d}}$ (or point forecast) offers only one value without uncertainty measure. In this case, the most likely solution with highest probability is chosen. However, most real life applications are subject to uncertainty, especially if human behaviour is involved in the system. In that case, based on the words of P. Saffo, the deterministic forecast represents the *illusory certainty*. Mathematically, one deterministic value without uncertainty has 100 % probability and infinite probability density. A more sophisticated way for prediction is the prediction interval (PI), which signifies the boundaries of possible values with a certain probability, in this example 95 %. The forecast including the most information is the *stochastic forecast* $\tilde{v}^{\mathrm{s}}$ providing the full predictive distribution, and thus the probability for each possible value of the forecast variable.

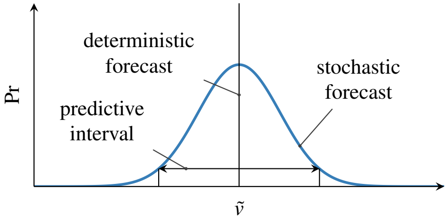


Figure 2.3: Illustration of the different forecasts that can be made, exemplary for future velocity $\tilde{v}$. The full stochastic forecast gives the probability for each possible value of $\tilde{v}$ and includes the most information. The prediction interval (PI) in this example indicates the min/max boundaries of 95 % of the values. The deterministic forecast predicts only one point and incorporates no information about uncertainty.

2.4 Validation and Evaluation Methods

With the methodologies explained in the previous section, forecast models for range estimation can be created. In general, an important aspect in modelling technical systems is their validation to prove that they are suitable for simulating system behaviour. A model can be validated using experimental measurements of the system by comparing the simulated quantities $\hat{x}$ to the measured result x, which can be viewed as the *true* or *desired* outcome. Whether a model is accurate enough can then be judged with suitable performance indices, also called error measures because they are based on the error $e = x - \hat{x}$. Commonly known examples are the mean absolute error (MAE) [139] and mean percentage error (MPE) [1]:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |x_i - \hat{x}_i| \tag{2.4}$$

$$\mathrm{MPE} = \frac{1}{n}\sum_{i=1}^{n} \frac{x_i - \hat{x}_i}{x_i} \; . \tag{2.5}$$

While validating a physical model with performance indices based on measurement data is a straightforward task, data-driven models such as machine-learning algorithms have to be analysed differently. Data-driven models are built using a part of the measurement data, so called *training data*. Calculating the model error with the same data set gives the *in-sample* error, which indicates how well a specific model can represent the given data. However, it is very important to check whether the built model also performs well on previously unseen data, the so called *test data*. These data are of the same nature as the training data but the values are unknown to the model. Evaluating the model error on the test data gives the *out-of-sample* error, which indicates the generalization performance of the model. Finding a general pattern instead of a pattern just specific to the chosen dataset is the goal of data-driven modeling. If a model shows good in-sample errors but high out-of-sample errors, it is called *over-fitted*. More information can be found in [47, 12].

To check both in-sample and out-of-sample performance of a data-driven model, *cross-validation* can be applied. The idea behind cross-validation is to split the overall available data into a training data set A and a test data set B. The model is trained with the data from A, afterwards the parameters are fixed and the performance on B is evaluated to get the out-of-sample error. Depending on the size and structure of the overall data-set, different kinds of cross-validation can be applied. For example, a *w*-fold cross-validation splits the total data-set into *w* parts and uses one of them for evaluating out-of-sample error and the rest for training. That process is repeated *w* times (folds) until all parts were used for evaluation once. An illustration of this process is shown in Figure 2.4.

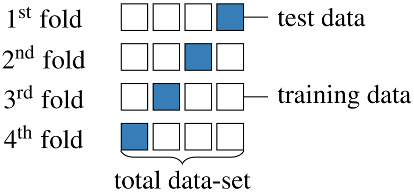


Figure 2.4: *w*-fold cross-validation. The total data-set is split into $w = 4$ parts and in each fold, one of the parts is used as test set and the rest for model training.

3 Range Estimation Methodology

This chapter introduces the overall range estimation methodology applied in this dissertation. It begins with the survey of related research in Section 3.1, which covers the general topic of range estimation and puts the presented dissertation into context. Following that, Section 3.2 illustrates the stochastic range estimation approach applied in this work. In that approach, stochastic range is expressed in terms of attainability. For that, a battery model is required which is introduced in Section 3.3. Apart from the models themselves, real test drive data is necessary to show and validate the developed algorithms. The method of obtaining the respective dataset is shown in Section 3.4. Finalizing the chapter, the post-processing of the collected data is explained in Section 3.5.

3.1 Survey of Related Research

Research interests in electric vehicles grew significantly in recent years. Key shortcomings of electric powered vehicles compared to conventional cars are the limited driving range and charging infrastructure. In consequence, drivers feel *range anxiety* which is the fear stranding with an empty battery. According to results from an electric vehicle field test study in [33], drivers reserve 20 % of the vehicle's range capacity as safety margin because they are afraid of stranding. One way to reduce range anxiety is a reliable range estimation algorithm which users can trust [26], hence a high safety margin is not necessary. In the following, this section covers the research activities in the respective field.

Fundamentally, driving range consists of two factors: energy consumption E_c and battery energy content E_b. Traditionally, the averaged past energy consumption is used as indication for future energy consumption. Taking the future trip into account, either from destination prediction [7, 70] or via user input, the energy consumption can be predicted with help of traffic and routing database (TRDB) information. This dissertation is focused on the reliable energy consumption prediction and uses a simplified model for the battery, presented in Section 3.3. Energy consumption prediction can be further divided into two categories: *microscopic* and *macroscopic* models following [69]. Microscopic models models rely on physical equations for predicting energy consumption whereas macroscopic models models use a set of explanatory variables to learn energy consumption directly from available data. Both approaches are introduced in the following.

Microscopic models represent the conventional engineering approach, where every factor influencing energy consumption is modelled explicitly. The system behaviour is comprehensible, therefore in system identification terminology, it is called *white-box* model. Creating a microscopic model to simulate the behaviour of electric vehicles using the fundamental equations and powertrain losses is a topic already covered in research, e.g. [46, 147, 8, 117, 128]. A microscopic model is illustrated in Figure 3.1. The necessary inputs are the predictions for future velocity $\tilde{v}$ and future acceleration $\tilde{a}$ as well as the static trip information trip length l and road gradient angle α. In addition, the driving resistance parameters $\mathbf{p}$ are required. Using these inputs and a physical model including driving resistance equations and powertrain model as illustrated in Section 2.1, the future power demand $\tilde{P}$ can be calculated and thus the future energy consumption $\tilde{E}_\mathrm{c}$. Mostly, microscopic models rely on fixed driving resistance parameters $\mathbf{p}$ and do not incorporate the inherent dynamics of these values [46, 147, 107, 69]. However, this assumption usually does not hold in a real-world application since for example, the vehicle mass m is not constant but relies on number of passengers or luggage just as the air density ρ is subject to weather conditions or altitude.

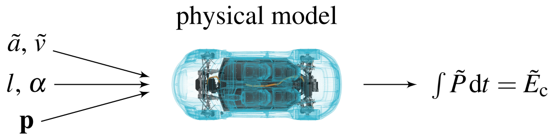


Figure 3.1: Microscopic model for energy consumption prediction, based on a physical representation of the vehicle. The dependency of the output $\tilde{E}_\mathrm{c}$ on the inputs $\tilde{a}, \tilde{v}, l, \alpha, \mathbf{p}$ is modelled using a white-box approach containing driving resistances and powertrain losses of the considered vehicle.

To track changing parameters, a suitable estimation algorithm can be included. Approaches for singular driving resistance parameters have been presented, for example using estimated rolling resistance for range calculation [143] or estimated vehicle mass in [30]. An overview about algorithms to estimate all entries of the vector $\mathbf{p}$ using standard vehicle sensors is given by [98]. These algorithms could be used to extend range estimation. However, looking at the required inputs for microscopic models in Figure 3.1, it can be seen that estimating $\mathbf{p}$ accurately just solves part of the problem, as $\mathbf{p}$ is just part of the inputs. Further variables are static TRDB information l, α, which pose no challenge as they are of geometric nature and can be downloaded after the driver has selected the destination. Yet, the future velocity $\tilde{v}$ and acceleration $\tilde{a}$ are not static and influenced by driver behaviour and traffic. Therefore, they must be predicted. Using a microscopic model for obtaining $\tilde{v}, \tilde{a}$ usually involves a deterministic driver behaviour model, as introduced in [65] but commonly the mean traffic speed $\bar{u}$ is used to derive the required inputs, e.g. [106]. Another approach would be applying a macroscopic prediction model [144] to provide the forecast to the microscopic energy consumption model.

Macroscopic models have a data-driven understanding of the system, hence physical and environmental influences are not modelled explicitly but instead they are implicitly learned from the data. Statistical machine learning methods are applied, which in recent years have grown more popular for a wide range of problems. The objective of the learning method is to find the *pattern* describing the influences on the output variable, in this case energy consumption [47]. Using the terminology of system identification, a macroscopic model is called a *black-box* model and is visualized in Figure 3.2. The input features π are obtained from a navigation database and contain the predictive information, e.g. $\bar{u}$, v_{lim}, α, l. Together with the measured output y, which is energy consumption, a mathematical model h is trained with help of a loss function $\Theta(\pi,y)$ [47]. After the model is trained, the future energy consumption $\tilde{E}_{\mathrm{c}}$ can be obtained using the future inputs $\tilde{\pi}$ and the identified model h. Examples for macroscopic models range from linear regression to neural network (NN) or deep learning applications [12].

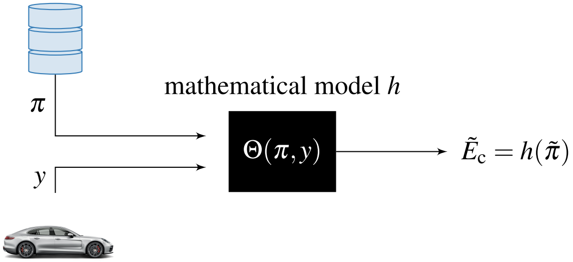


Figure 3.2: Macroscopic model for energy consumption prediction, based on a mathematical representation of the dependencies. The available input data π together with the desired output data y are used to train a black-box model h using a suitable loss function Θ. The trained model can subsequently be used to predict future values.

[77] presented a macroscopic approach, where a neural network (NN) is used to predict energy consumption based solely on the individual driving data. Those individual features can be extended by cloud-data from more than one vehicle, a method used by [148]. In that work, the used machine learning algorithm is also based on a NN structure. Naturally, other learning algorithms are possible, such as the popular support vector machine (SVM) used by [42] to predict energy consumption from crowd-sourced data. Apart

from such complex model stuctures, [102] proposed a multi-linear regression (MLR) with a very small set of features, which shows good performance in a simulation environment.

For both model categories, micro- and macroscopic, various research articles exist and only a selection was mentioned above. However, most of the publications only offer deterministic solutions to the problem of range estimation. The variety of influences on electric vehicle range already makes an exact solution only a theoretical possibility. As introduced in Section 1.2, the stochastic nature of the system should be considered, especially because human behaviour has a high impact on future energy consumption and cannot be determined perfectly. While this aspect is recognized in some related work [27, 146], often it is not incorporated in the prediction. In contrast, many publications aim at including a high number of inputs (features) into the algorithms, which is a problem also pointed out by [63]:

> Other researchers have been aware of the problems arising from inaccurate predictions of resources in cars and other battery-dependent technologies. However, most of this prior work has focused on designing algorithms that improve prediction accuracy by increasing the number of parameters that the system takes into account. [63, p. 3]

That means, instead of dealing with all possible input factors as algorithm features, modelling the system's uncertainties would result in a better prediction because it would be more *reliable*, as illustrated in Section 1.2. In their work, Jung and Steinert [63] illustrated that range anxiety of BEV drivers can be decreased by including an uncertainty indication in the range display. Still, this was a psychological study and the method of how to obtain the uncertainty was not covered. Following the idea of including uncertainty in the range estimation, Oliva et al. [84] presented an algorithmic approach for obtaining the prediction interval (PI). They demonstrated this method in a simulation using standard driving cycles. Furthermore, Ondruska and Posner [86] introduced a computationally efficient algorithm to account for range uncertainty and implemented it in a test vehicle. However, research on stochastic forecasting for the specific problem of range estimation is very limited. Instead, a lot of papers exist where the deterministic range estimate is used for assistance systems such as charge-planning, e.g. [9, 14, 18, 72, 137]. This dissertation aims at helping to fill this research gap by developing a stochastic forecasting methodology with respect to individual driving behaviour and vehicle configuration. The resulting reliable forecast could then be used for range display as well as assistance systems and thus decrease driver's range anxiety. A pilot study on this subject illustrating the problem has already been presented to the research community in [108].

3.2 Proposed Approach for Range Estimation

The estimation of electric vehicle range depends on the energy consumption forecast $\tilde{E}_\mathrm{c}$ and the estimated battery energy content $\hat{E}_\mathrm{b}$. Traditionally, driving range is expressed in terms of the distance until the vehicle runs out of energy, called *distance-to-empty* [102]. The driver consequently compares the range value to the trip distance in a binary assessment whether a destination is attainable. In this dissertation, the uncertainty of both energy consumption and energy content shall be included, therefore they are modelled as random variables. Consequently, attainability p_a can be calculated directly, indicating the probability of reaching a selected destination for a given route and battery condition [86]:

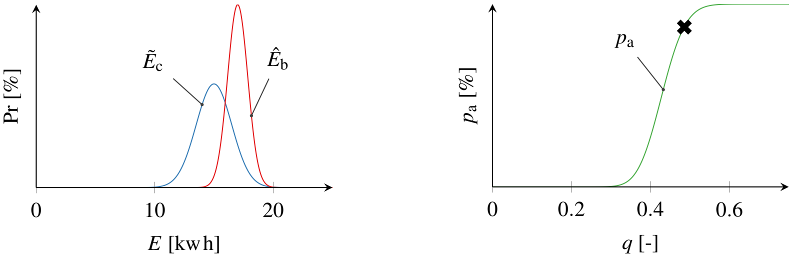
$$p_\mathrm{a} = \Pr(\tilde{E}_\mathrm{c} \leq \hat{E}_\mathrm{b})\,. \tag{3.1}$$

In the following, attainability p_a serves as stochastic interpretation of driving range. Thereby, (3.1) also works for the deterministic case and covers the traditional approach.

It is also possible to calculate attainability not only based on the current battery state of charge (SOC) but for all possible SOC values. Hence, if attainability is too low to take the trip safely with the current SOC, the question of *how much SOC is required to reach the destination safely* can be answered. In practice, with the knowledge of the difference between current and required SOC q, the amount of charging energy can be obtained and thus, the amount of charging time at a specific charging station. An example is shown in Figure 3.3, where Figure 3.3a illustrates a specific situation with $\tilde{E}_\mathrm{c} \sim \mathcal{N}(\mu = 15\,\mathrm{kWh}, \sigma = 1.5\,\mathrm{kWh})$ and $\hat{E}_\mathrm{b} \sim \mathcal{N}(\mu = 17\,\mathrm{kWh}, \sigma = 0.85\,\mathrm{kWh})$. Using (3.1), resulting attainability in this case is $p_\mathrm{a} = 87\,\%$. This point is illustrated with the black cross in Figure 3.3b, where p_a over the whole SOC range q is shown. With the resulting curve, required SOC for other attainability values can be obtained, e.g. for $p_\mathrm{a} = 99\,\%$. Taking a charge-planning scenario, the proposed methodology therefore provides vital information: firstly, how likely can the charging stations on the route be reached? Secondly, if recharging is required, how long does it take at the different charging stations? With this information, an optimization algorithm could find the best possible strategy for the trip.

For the driver, this methodology also offers advantages. Since safe arrival at the destination can be calculated incorporating uncertainties directly, constant safety margins such as the 20 % found by [33] become unnecessary. On the one hand, this allows using the full potential of the battery and on the other hand prevents stranding. That raises acceptance of electric mobility, saves energy and decreases range anxiety. An additional opportunity is to display range uncertainty to the driver, which reduces stress levels according to [63].

After presenting the definition for stochastic range calculation, subsequently the approach for obtaining $\tilde{E}_\mathrm{c}$ and $\hat{E}_\mathrm{b}$ must be designed. Thereby, the focus of this work lies on the energy consumption forecast (ECF) whereas for battery content estimation, a simplified model is used which is presented in Section 3.3. An overview can be seen in Figure 3.4. The parameter esti-



a) Energy consumption $\tilde{E}_\mathrm{c}$ and battery energy content $\hat{E}_\mathrm{b}$ expressed as random variables.

b) Destination attainability p_a over the whole SOC range.

Figure 3.3: Calculating attainability as stochastic expression of electric vehicle range. Figure 3.3a shows energy content and energy consumption as random variables. Energy content $\hat{E}_\mathrm{b}$ has lower uncertainty than $\tilde{E}_\mathrm{c}$ which can be seen by the lower standard deviation. Calculating attainability p_a from these values results in the black point illustrated in Figure 3.3b. The green curve represents attainability over the whole SOC range q with fixed energy consumption.

mation (PE) algorithm obtains $\hat{\mathbf{p}}$ using vehicle sensors and a recursive filter illustrated in Section 4.1. Apart from the driving resistance parameters, the future velocity profile $\tilde{v}$ is required for an ECF, which is provided by the velocity prediction (VP) method introduced in Section 5.1. The VP method in this dissertation is a macroscopic model learning driving behaviour from the data. Thereby, the surrounding traffic situation must be assessed in order to learn contextual driving behaviour. Intuitively, the algorithm should be able to detect a jammed traffic situation since there, only car following behaviour can be observed. Therefore, the traffic phase classification (TPC) shown in Section 4.2 generates an estimate for traffic phase $\hat{\phi}$ using the appropriate sensors. Subsequently, $\hat{\phi}$ is used as part of the input features π for the macroscopic VP model. Together, $\tilde{v}$ and $\hat{\mathbf{p}}$ are used for the ECF based on a microscopic powertrain model which is explained in Section 5.2. Afterwards, the resulting $\tilde{E}_\mathrm{c}$ can be compared to the battery energy content estimate $\hat{E}_\mathrm{b}$ from the battery model for calculating attainability p_a according to (3.1). The acquisition of required test drive data for the data-driven algorithms within the presented methodology can be seen in Section 3.4.

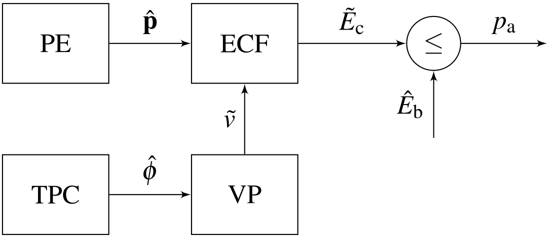


Figure 3.4: Overview of the approach for calculating attainability using the individual algorithms: parameter estimation (PE), traffic phase classification (TPC), velocity prediction (VP) and energy consumption forecast (ECF). The resulting $\tilde{E}_\mathrm{c}$ is compared to the battery energy content from the battery model $\hat{E}_\mathrm{b}$ to obtain attainability p_a.

3.3 Battery Model

The range estimation method presented in the previous section requires a battery model for calculating $\hat{E}_\mathrm{b}$. The model used in this dissertation is derived from [108] and presented in this section. The basic problem is that the battery's energy content E_b cannot be measured directly and a number of steps and measurements have to be performed for its determination. Since all measurements are prone to uncertainties, *measurement noise*, and obtaining $\hat{E}_\mathrm{b}$ involves various measurements, the total resulting uncertainty is likely to be significant. Even the use of sophisticated observer models, adaptive filtering techniques or machine learning methods cannot eliminate this uncertainty completely [127, 16].Therefore, E_b is considered a random variable in this dissertation. An overview of battery state estimation techniques can be found in literature, e.g [73, 74, 141].

Obtaining the energy content of the battery is conventionally based on the measurement of state of charge (SOC). Uncertainty in SOC measurement mainly result from uncertain voltage measurement in each cell. [56] proposed an algorithm with around 2 % SOC measurement error, while current implementations in modern day production cars have larger errors. Aside from the SOC, energy content does also depend strongly on battery age. The capacity fades over lifetime and is measured in state of health (SOH), the percentage of actual capacity from the original capacity. Accuracy of SOH estimation ranges from 80 % to 98 % according to [11]. In addition, the usable energy of the battery also depends on the discharge profile because internal losses of the battery increase with rising load and therefore battery efficiency is affected. That gives a feedback loop, as the future load profile is also uncertain.

In this dissertation, the characteristics of the battery are not the focus and therefore a simple model is used. It is based on an equivalent circuit model comprised of an ideal voltage source and an ohmic resistance, see Figure 3.5. There, $U_0(q)$ is the open circuit voltage, dependent on current SOC q, R_i is the internal resistance, P_{bat} is the battery power, U the resulting clamping voltage and I_{bat} the battery current. Environmental conditions, such as the dependency of the battery states on the temperature, as well as complex internal states, such as the dependency of the losses on the exact load profile are neglected. The latter, battery efficiency change dependent on the discharge profile, was shown to be much less of an influence than SOC and SOH measurement errors in [108]. For these two, moderate uncertainties of 3 % for SOC and 5 % for SOH are assumed, which is in accordance with the literature.

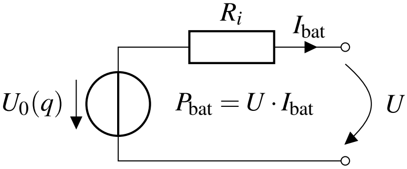


Figure 3.5: Equivalent circuit model of the battery.

3.4 Test Drive Data

The methodology presented in Section 3.2 comprises data-driven algorithms and models. Therefore, real test-drive data is required for their development which can also be used to prove the validity of the proposed methodology in a real application and not simply based on simulations. For data acquisition, two series of test runs are executed, one with a battery-electric vehicle (BEV) comprising 380 km and one with an internal combustion engine vehicle (ICEV) comprising 1900 km. The BEV is a full-electric PORSCHE BOXSTER with four-wheel drive and a typical range of around 150 km (Figure 3.6). The front axle is powered by a permanent magnet synchronous motor with 120 kW and the rear axle is powered by an 140 kW asynchronous motor [10]. The high-voltage battery has a peak power of 270 kW and a nominal capacity of 35 kWh. For this work, a global positioning system (GPS) module and an online connection was installed to download real-time information from the traffic and routing database (TRDB) via an application programming interface (API) request. The BEV data is used for validating the powertrain model presented in Section 4.1 and is is gathered on a random set of test drives around Weissach in southern Germany.

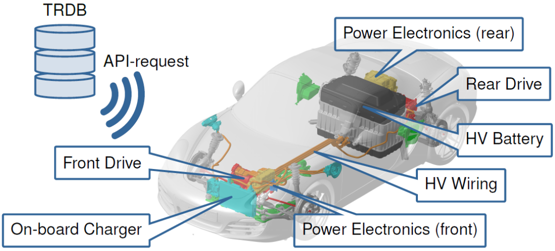


Figure 3.6: Battery Electric Vehicle (BEV) used for the test drives. The vehicle topology comprises a four wheel drive system with two electric motors. Via the online-connection and the GPS antenna, it is able to download map and traffic information.

The BEV has a central limitation in its usefulness for the algorithms presented in this dissertation, which is the absence of camera and radar. Without that, other traffic participants, road signs, lane markings etc. cannot be detected and the local traffic situation could not be obtained by the TPC algorithm. Therefore, a vehicle with the required sensors is used to collect the data, the PORSCHE PANAMERA, which is an internal combustion engine vehicle (ICEV), see Figure 3.7. By using the validated powertrain model of the BEV, the equivalent electric energy consumption E_c can be simulated for the ICEV dataset. Thus, all required sensor information can be gathered but still a validated electric vehicle powertrain model is used, therefore the energy consumption behaviour of the test runs is realistic. In addition, since a powertrain model is used, the considered environmental influences can directly be defined and methodical bias can be excluded. Consequently though, not directly specified influences are not considered. Until properly equipped electric vehicles are available, this is a practical approach.

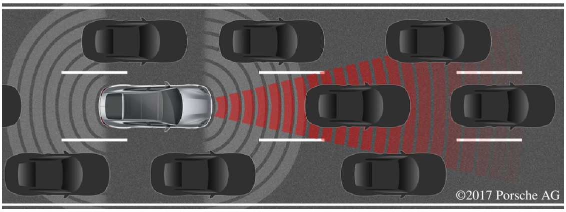
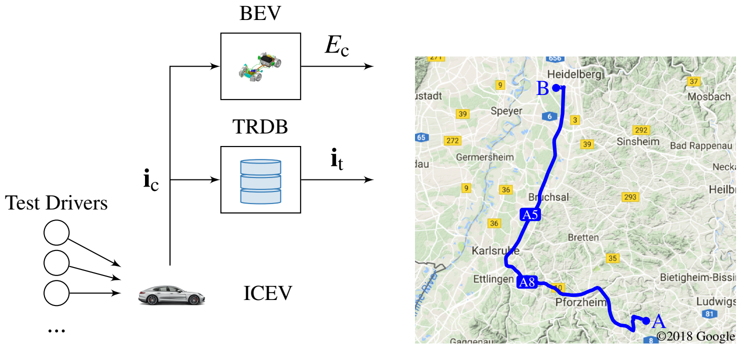


Figure 3.7: Internal Combustion Engine Vehicle (ICEV) used for the test drives. Via the installed camera and radar technology, signs, other vehicles and lanes can be detected.

Since the ICEV dataset is used for algorithm training and thus for modelling driving behaviour, it is not gathered randomly but follows a specified approach, which is illustrated in Figure 3.8. A series of test drives with 10 different drivers was executed on a specified round trip which mainly contained highway roads between Weissach and Heidelberg in southern Germany (see Figure 3.8b). The test drives were carried out on different times of day to catch different traffic patterns, e.g. in the morning rush or during low traffic density phases around noon. Each driver was told to drive equal to their normal driving style and no other specifications were given. Since the test drives were executed with the ICEV, influences of range anxiety on driving style can be excluded.



a) test run method

b) drive route

Figure 3.8: Illustration of the test runs. Figure 3.8a shows an overview of the method: The different test run participants drive the specified route with the ICEV. The controller area network (CAN) data $\mathbf{i}_c$ is used to download the TRDB data $\mathbf{i}_t$ and to simulate equivalent electric energy consumption E_c using the BEV powertrain model. The test drive route between Weissach (A) and Heidelberg (B) is shown in Figure 3.8b. The test drivers did a round-trip A-B-A, which is mostly on the highways A8 and A5.

All available vehicle states were measured with the standard CAN system of PORSCHE production cars at 100 Hz and measured CAN-data $\mathbf{i}_c$ was transferred to the powertrain model to simulate the equivalent electric energy consumption E_c. Based on the GPS and time signal, the information from the TRDB $\mathbf{i}_t$ was downloaded for the whole trip which contains details about the selected route by the provider HERE [48], as introduced in Section 2.2. The velocity profiles of all test drivers are plotted in Figure 3.9. There, the overall variance of driving speeds due to differences in traffic state and driving style can be observed.

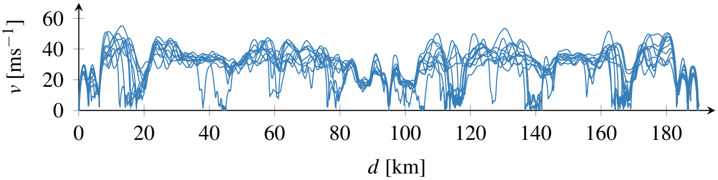


Figure 3.9: Velocity profile of the 10 test drivers for the round-trip.

When using ICEV test data to derive conclusions for BEV range estimation algorithms, it is assumed that driving behaviour is comparable for BEVs and ICEVs. To confirm that, the test drive data can be used. Thus, a random subset of the ICEV data with different drivers and the same total length as the BEV data is chosen. It is suspected that due to the different torque characteristics, the acceleration behaviour of the BEV differs from the ICEV. The statistical feature selected to describe acceleration behaviour is the standard deviation σ_a. In Table 3.1, it can be seen that it does not differ significantly although σ_a of the BEV is a little higher than the ICEV value as expected. For additional analysis, the findings of Ericsson et al. [29] are applied where statistical features are used to describe driving style. In that work, the most important feature to describe driving style and and its effect on energy consumption is the relative positive acceleration (RPA). Therefore, it was also extracted from the data sets and the results are shown in Table 3.1. Again, the values do not differ significantly with the RPA of the BEV data being a little higher. While this simplified comparison is not complete, and more research should be done in this area, it shows that the driving style is not significantly different and therefore the statements made about range prediction from this method are valid. In fact, it is one of the goals in automotive development to provide the same driving characteristics in a BEV compared with the ICEV most drivers are used to. This methodology was already published to the research community in [111].

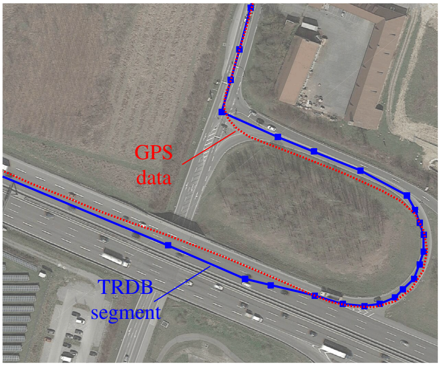
Table 3.1: Driving behaviour comparison.

	BEV	ICE
distance [km]	380 km	380 km
σ_a [m s^{-2}]	0.6	0.5
RPA [m s^{-3}]	3.1	3.0

3.5 Post-processing of Measurement Data

The collected data from the test drives presented in the previous section must be post-processed in order to be usable for the range estimation algorithm, e.g. due to measurement noise. However, one of the most important aspects is the matching of GPS-based $\mathbf{i}_\mathrm{c}$ data onto the map of the TRDB to merge CAN signals with map and traffic information correctly. Thereby, the map is built using segments k and their individual properties, with the GPS shape among them. To illustrate the problem, Figure 3.10 shows both the measured GPS signal from the CAN and the map segments from the TRDB on a highway on-ramp of one test drive. It can be seen that the TRDB segments cut the road into various parts, and the segment length differs depending on the position, for example there are more and shorter segments in the curve. Map matching comes down to finding the map segment closest to the measured GPS signal from the CAN. If the closest segment is found, the data can be merged, e.g. the map information v_lim can be compared with the driven speed v. To achieve adequate matching accuracy, first the map segments are split into smaller parts of 10 m, which in turn is the maximum possible error. With a *k-nearest-neighbour* algorithm, the closest segment part to every measurement point is identified. An explanation of this algorithm can be found in [47].

Having obtained the correct corresponding segment allows to assign the TRDB properties to each data point. One of those properties is the probe speed distribution u in a 5 min resolution. Some segments may have no

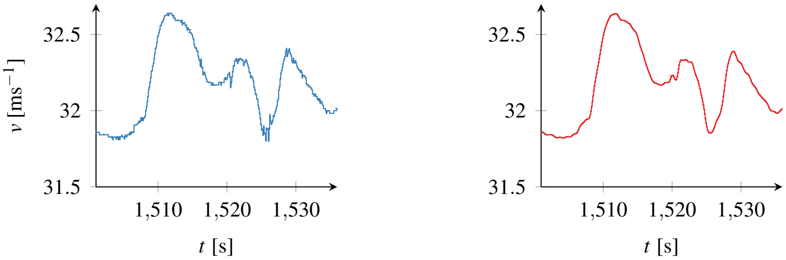


©2018 Google

Figure 3.10: Map matching problem for the test drives. The map segments are illustrated in blue, the squares denote the segment start and end points. The measured GPS data points from the CAN are depicted in red. The task of map matching is to find the corresponding segment for each measurement point.

probe information for the specific spatio-temporal query, either because coverage was too low or nobody drove on that segment at the time. In that case, it can be assumed that there is no congested traffic phase and the typical mean free-flow speed $\bar{u}_\mathrm{f}$ is used. In this work, $\bar{u}_\mathrm{f}$ is obtained by analysing the probe speeds for the specific segment between 0:00 and 5:00 AM over the course of the last month which is similar to the approach of [48]. Implicitly, it is assumed that in this time period, no congested traffic occurs.

Apart from the map-matching problem, the CAN data $\mathbf{i}_\mathrm{c}$ also needs to be post-processed because the sensors generating the signals are subject to measurement noise. Measurement noise describes the distortions and disturbances or discretization errors coming along with the digital signal from any measurement device installed in a real-world application. To reduce measurement noise, signal processing methods are applied [133]. In this dissertation, low-pass filters with different settings depending on the signal are used for noise reduction. For example, the velocity signal v originat-



a) original signal

b) filtered signal

Figure 3.11: Illustration of the noise in the velocity signal. The original data from the sensor is illustrated in Figure 3.11a. The same signal smoothed by a lowpass filter without phase shift is shown in Figure 3.11b.

ing from the vehicle's electronic stability program (ESP) is filtered with a 5th order infinite impulse response (IIR) lowpass filter. The passband frequency is set to 1 Hz and the passband ripple to 0.1 Hz. Since the signal is post-processed and not filtered in real-time, the inherent phase shift of the filter can be subtracted. The result of this noise reduction method can be seen in Figure 3.11, where the filtered velocity signal is smoother than the raw value. Although the level of noise seems to be low in this example, it could result in significant errors if algorithms use the unfiltered signal. For example, if the derivative of the noisy velocity signal $\mathrm{d}v/\mathrm{d}t$ is calculated to obtain acceleration a, even small amounts of noise can have a high impact on the system. For more details on signal filtering theory, see [133].

4 Estimation Algorithms for Predicting Energy Consumption

The stochastic range estimation method illustrated in Section 3.2 requires data-driven estimation algorithms, namely parameter estimation (PE) and traffic phase classification (TPC). These algorithms are explained in the following chapter.

At first, Section 4.1 illustrates the microscopic energy consumption model used in this dissertation. For this model to be accurate, the driving resistance parameter vector **p** must be estimated, which is the task of the PE algorithm shown in Section 4.1.1. The resulting model must be extended for a forecast application since the quality of predictive information differs from the measurement data available for estimation. Section 4.1.2 explains the required extension possibilities before the performance of the applied models is analysed in Section 4.1.3. Thereby, special consideration is given to the use of robust performance indices.

The second part of the chapter, Section 4.2, introduces the traffic phase classification (TPC) algorithm, which is vital for recognizing the local traffic situation around the ego-vehicle. TPC is based on a pre-defined feature set, which is illustrated in Section 4.2.1. Based on those features, different approaches for classification algorithms are created and presented in Section 4.2.2. The performance of those different algorithms is subsequently compared in Section 4.2.3.

4.1 Energy Consumption Model using Parameter Estimation

Tractive energy consumption of an electric vehicle can be calculated based on the driving resistances and powertrain losses, as illustrated in Section 2.1. There, the powertrain model of the BEV used in this dissertation was already introduced, which calculates energy consumption based on the driving resistances F_r and the vehicle velocity v. In turn, F_r can be obtained based on the driving resistance equation (2.1) with the short version

$$F_\mathrm{r} = \mathbf{p}^\top \cdot \mathbf{B} \tag{4.1}$$

where $\mathbf{p}$ is the parameter vector and $\mathbf{B}$ the measurement vector. It can be seen that $\mathbf{p}$ is an important factor influencing energy consumption and comprises the following values:

$$\mathbf{p} = \begin{pmatrix} \underbrace{mf_\mathrm{r}}_{p_1} & \underbrace{1/2\,c_\mathrm{w}\rho A}_{p_2} & \underbrace{m}_{p_3} \end{pmatrix}^\top . \tag{4.2}$$

Based on the vehicle topology of the battery-electric vehicle (BEV) presented in Section 3.4, their individual influences on total E_c can be analysed and the results are shown in Figure 4.1. It can be seen that m has the highest influence, a 10 % change leads to a 5 % energy consumption deviation. The amount of mass change is realistic for the BEV used as test vehicle in this dissertation, in fact it can change up to 10.1 % depending on luggage and passengers. However, it is a small sports car, for the electric SUV TESLA MODEL X, mass can increase by more than 20 % [124]. Electric buses or vans have even more potential for mass change [31]. The second largest influence is induced by a change in the rolling resistance coefficient. Changes in f_r result from differences in tire characteristics (tread pattern, temperature) and road conditions, even 100 % change is possible between asphalt and tarmac according to [80]. Finally, also the aerodynamic drag coefficient c_w and air density ρ cause a deviation in E_c. Thereby, c_w has an uncertainty of up to 10 % according to wind tunnel tests performed in [23]. ρ is dependent on multiple factors such as elevation and outside temperature. Based on the equations explained in [93], the following can be calculated: From −5 °C to 30 °C, air density decreases by 11.5 % and from an elevation of 0 m to 1000 m, it decreases by 12 %. Further derivations originate from wind speed and direction. As introduced in Section 3.1, a lot of publications about energy consumption models for range estimation rely on constant parameters from test bench measurements. However, based on the presented numbers in in Figure 4.1, it is imaginable that an approach using constant parameters can lead to significant bias and uncertainty. Therefore, in this work the parameters are estimated using the algorithm presented in the next section.

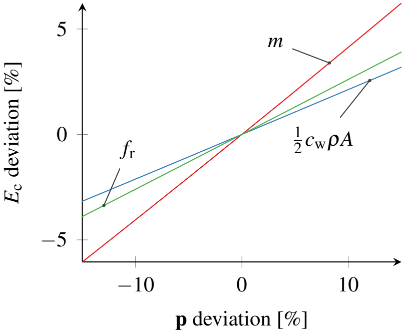


Figure 4.1: Sensitivity of the energy consumption dependent on the changes of the individual driving resistance parameters. Basis for the results is the vehicle topology of the BEV. m shows the highest influence on energy consumption with a 10 % change leading to 5 % E_c deviation.

4.1.1 Parameter Estimation Algorithm

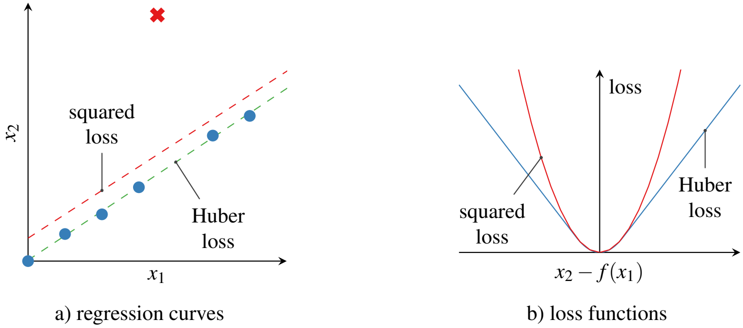
Since the driving resistance parameters $\mathbf{p}$ can change significantly and this has a high impact on energy consumption, they need to be estimated in the vehicle during driving operation. This is the task of the parameter estimation (PE) algorithm presented in this section, which has already been presented to the research community in [109].

The basic equation used for the estimation problem is (4.1). There, $\mathbf{B}$ can be measured directly, F_r can be obtained with the powertrain model introduced in Section 2.1 and $\mathbf{p}$ must be estimated. This mathematical problem is linear in the parameters, therefore it is a multiple-input-single-output (MISO) system. This type of system and various algorithmic applications on PE have been studied extensively in [98] and hence only an overview is given in this section. A widely studied solution for the MISO problem is least-squares (LS) which leads to optimal results for non-biased normally distributed residuals r. The LS loss function is given by:

$$\min_{\hat{\mathbf{p}}}\left[\sum_t r_t^2\right] = \min_{\hat{\mathbf{p}}}\left[\sum_t (F_{\mathrm{r},t} - \hat{\mathbf{p}}_t \mathbf{B}_t)^2\right] \tag{4.3}$$

where $\hat{\mathbf{p}}$ denotes the estimated parameters and t the individual considered time-steps. The mathematical formulation in (4.3) is called a *batch* solution, where all the data is computed at the same time. However, for an online application in the vehicle, the batch formulation must be turned into a *recursive* formulation since the data is obtained sequentially. This results in the commonly known recursive-least-squares (RLS) algorithm, which was used to estimate $\mathbf{p}$ as shown in [99]. However, it is stated also that RLS fails obtaining an accurate solution in the presence of outliers. This makes its use infeasible in many applications, as outliers will occur naturally within a real-life environment.

Estimation algorithms which are able to deal with outliers are called *robust* estimators. A commonly known family of robust estimators is the M-estimators introduced by [51]. M-estimators have an altered loss function Θ, such as the Huber loss function from [52], which imposes less weight on values far from the regression curve, thus decreasing the effect of outliers. The LS function in (4.3) represents a special case where $\Theta = (\cdot)^2$. Changing the LS loss function to another Θ turns RLS into the weighted-least-squares (WLS) algorithm, which was shown in [100]. In Figure 4.2, an exemplary comparison of the LS and the Huber loss function is shown. There, Figure 4.2a shows some data points with zero mean Gaussian noise of low variance in blue over the dimensions x_1, x_2. One outlier is induced, signified with the red cross. When linear regression with the standard LS loss function is performed on these points, the red regression curve is obtained which has significant bias. When using the Huber loss function, the outlier is less important and thus the regression curve is less biased. The characteristics of LS and Huber loss are depicted in Figure 4.2b based on the difference between regression curve $f(x_1)$ and the data points. It can be seen that less weight is put on outliers using the Huber loss compared to LS which makes this solution robust.



a) regression curves

b) loss functions

Figure 4.2: Problem of robustness for regression-based estimation. In Figure 4.2a, multiple data points for regression are displayed in blue, with one outlier in red. The regression curves for the Huber loss and the squared loss are illustrated. Figure 4.2b shows the loss functions themselves and their difference.

Another problem of online estimators occurs if the system is subjected to poor excitation. In a vehicle, this can easily happen, e.g. if the velocity remains constant on the highway over a certain period of time. The data points used for the inherent regression of the estimation algorithm lose variability and the solution becomes unstable. Figure 4.3 illustrates this problem which could occur for regression-based estimators. Figure 4.3a shows the profile of a system input variable x_1 and an output variable x_2 over time. From the time-based profile, it can be seen that the system experiences poor excitation. When performing regression on these data, the points show marginal differences, as illustrated in Figure 4.3b. A regression would be unstable because essentially, it is performed on only one point. For recursive algorithms, that means a substantial growth of the covariance. This problem is called *windup problem* and it can be prevented using *regularization* methods shown in [98]. An alternative approach was shown by [121], where an altered version of the Kalman filter [64] was presented. Essentially, it limits the growth of the covariance and therefore prevents unstable solutions. The performance of this algorithm in its robust version is compared to the standard Kalman filter in [100], and its superiority was shown using real test drives. For more information about the mathematical theory behind robustness and wind-up stability, see the work of [98] since in the following, known implementations of the algorithms are applied.

Although the Kalman filter introduced by Stenlund and Gustafsson [121] solves the robustness and windup problem, there is still the trade-off between flexibility and stability, which poses a challenge for recursive estimators. That means an algorithm which is able to track parameters well and is robust in the presence of outliers could fail to detect sudden changes in the system. Essentially, the robust algorithm would treat the data points of a suddenly changed system as outliers and would need a long time to converge to the true value again. A way to deal with this trade-off is using multiple model estimation (MME) [119]. The idea is to use several estimators in-parallel with different noise settings. The estimator with the lowest

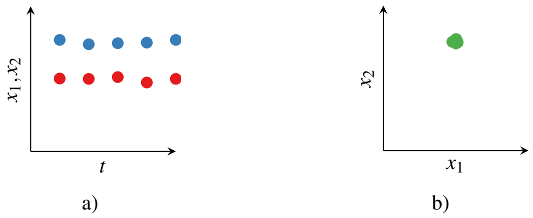


Figure 4.3: Problem of wind-up for regression-based estimation. Figure 4.3a shows an input variable x_1 and an output variable x_2 over time. Because there is poor excitation in the profile, the regression points in Figure 4.3b have marginal differences. This causes unstable solutions.

residuals over a certain period of time offers the best solution for the system behaviour and its parameter estimations are the most accurate. Thereby, in standard applications those different estimators are always running, therefore the computation time is significantly and permanently increased compared to using only one estimator.

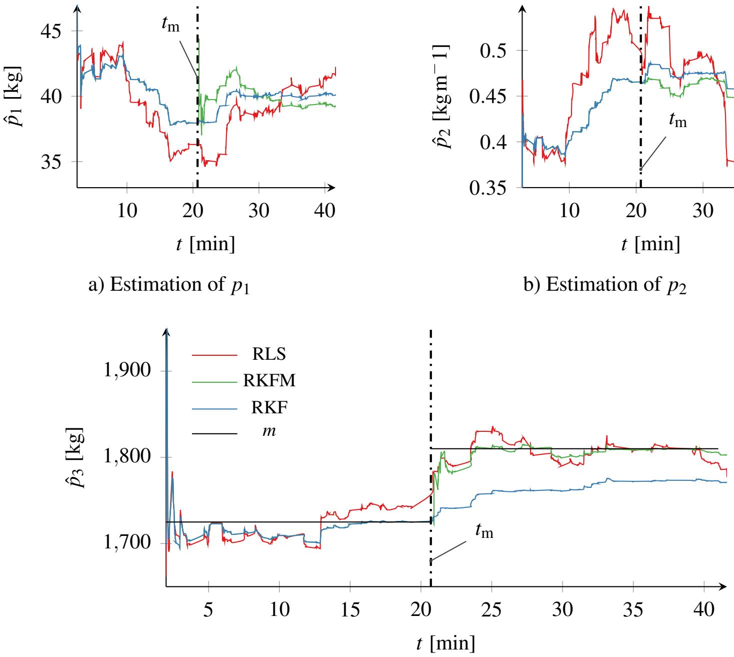
Hence, in this dissertation, an event-based MME is shown which has been published previously in [109]. Two models are used, (M_1 and M_2). In normal driving situations only M_1 is computed. Model M_2 is more sensitive and only activated, when there is a possibility that parameters change, e.g. a standstill of the vehicle for $\geq 3\,\mathrm{s}$ where someone could enter or leave the vehicle thus changing the mass. Thereby, the both models have the same parametrization with the exception of the covariance matrix. M_2 has significantly higher values to account for increased sensitivity. Subsequently the residuals of the converged model M_1 and the sensitive model M_2 show whether the system has changed or not. Therefore, the probability for each model M_j being the correct model is calculated after meeting the standstill condition:

$$\Pr(M_\mathrm{j}|\bar{r}_{i,j}) = 1 - \frac{|\bar{r}_{i,j}|}{|\bar{r}_{i,1}| + |\bar{r}_{i,2}|}\,. \tag{4.4}$$

$\bar{r}_{i,j}$ is the averaged residuals from the time standstill ended to timestep i for model j, where Δi is 10 ms. $\Pr(M_\mathrm{j})$ is calculated for 2000 timesteps i after standstill where higher probability means the model has smaller residuals. Then, the model with a better fit and therefore reduced estimation error can be selected.

To show the differences between the presented estimator alternatives, real driving tests are executed with the BEV. For highlighting the introduced problems from above, a test drive with mass change is presented here, since it is directly measurable and controllable. The vehicle mass with one person in the car is obtained and the test drive begins with this fixed setting. After 20.7 min until t_m, a second person entered the car, increasing the mass which again can be measured. The task of the estimation algorithms is to track this change accurately. Candidate algorithms are the standard, widely used recursive-least-squares (RLS) algorithm, the robust and windup-stable Kalman filter (RKF) of [121] and the extension, RKF with MME (RKFM). All tuning parameters are defined following [2], therefore they are not mentioned in this dissertation. Since the value of the tuning parameters are strongly dependent on the specific data of the problem, it is recommended to follow the tuning guidelines of [98] for similar implementations. There, one can find MATLAB algorithms for each filter to download.

Figure 4.4 shows the estimation of the driving resistance parameter vector $\mathbf{p}$, divided into the individual entries p_i following (4.2). Figure 4.4a and Figure 4.4b show the estimates for the parameters of rolling resistance and aerodynamic resistance which are not constant during a trip and cannot be measured exactly, e.g. due to unknown ρ. Therefore, the evaluation is concentrated on $\hat{p}_3$, which directly represents mass estimation $\hat{m}$ (Figure 4.4c). Before t_m, both RLS and RKF stay within 2 % around the true value m. RLS does not converge due to the above mentioned robustness and windup problems. The filter addressing these issues, RKF shows better performance and converges to m accurately. That means this filter is appropriate for a vehicle implementation and deals with real-life problems accordingly. However,



a) Estimation of p_1

b) Estimation of p_2

c) Measured mass m and estimation of p_3

Figure 4.4: Parameter Estimates with the presented algorithms for the test run with mass change. Until t_{m}, only the driver is inside the vehicle and at t_{m}, a second person enters the car. The accuracy of $\hat{p}_1$ and $\hat{p}_2$ cannot be evaluated exactly, whereas the true mass can be measured. Therefore, by the tracking performance of $\hat{p}_3$, the different algorithms can be compared.

after t_m, RKF fails to track the value because it treats the data points from the changed system as outliers. This is where the advantages of MME can be used. At t_m, the standstill condition is met and therefore M_2 is activated and the probability for each model is calculated using (4.4). Thereby, the probability of the more sensitive model M_2 is higher than the converged model M_1, indicating a system change. In consequence, RKFM switches to the estimator M_2. Because of that model switch, the RKFM can track the true value after t_m, thus illustrating the usefulness of MME in this application. Therefore, in the following of this dissertation, the driving resistance parameters are estimated using RKFM for an accurate energy consumption model. Thereby, the important fact for range estimation applications is the performance of the complete energy consumption, where PE is a part of. This is analysed in Section 4.1.3.

4.1.2 Predictive Model Formulation

In the previous section, the PE algorithm was introduced to achieve accurate estimates $\hat{\mathbf{p}}$ and thus an accurate energy consumption model. In theory, this model could also be used to obtain the forecast $\tilde{E}_\mathrm{c}$. Therein, the future driving resistance force $\tilde{F}_{\mathrm{r},k}$ on each route segment k can be calculated based on the driving speed forecast $\tilde{v}_k$ and average gradient $\bar{\alpha}_k$ using the route information from the TRDB:

$$\tilde{F}_{\mathrm{r},k} = \hat{\mathbf{p}} \cdot \begin{pmatrix} g \cos \bar{\alpha}_k \\ \tilde{v}_k^2 \\ g \sin \bar{\alpha}_k \end{pmatrix}^\top \tag{4.5}$$

Note that the term including acceleration resistance is missing because no prior information on future vehicle acceleration $\tilde{a}$ can be obtained from TRDB data. That lies in the nature of the data structure since the speed information therein comprises aggregated probe information over the segment length l_k, e.g. mean traffic speed $\bar{u}_k$. From that, the exact future velocity profile is not available even for ideal and unbiased input data, a problem also pointed out by [69]. To illustrate this, the different velocity information for an exemplary test drive is shown in Figure 4.5. The measured velocity v can be seen in comparison to the mean traffic speed over the segments $\bar{u}$ from TRDB. The ideal deterministic velocity forecast would yield the averaged velocity $\bar{v}_k$, as it is the true value. Using this forecast together with (4.5) directly for obtaining future energy consumption is denoted $\tilde{E}^{\mathrm{d}}_{c|a=0}$, as it is a deterministic forecast without considering acceleration force. It can be imagined that such a model could result in biased predictions, therefore it needs to be extended for predictive applications.

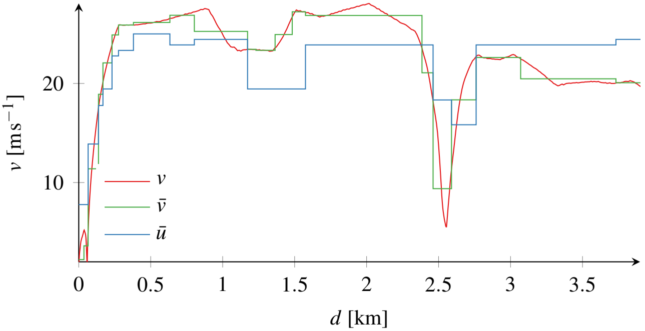


Figure 4.5: Comparison of measured vehicle velocity v, averaged velocity $\bar{v}$ over the segment length l_k and traffic speed $\bar{u}$ from TRDB for an exemplary test drive.

In consequence, researchers often induce constant acceleration between the deterministic predictions $\tilde{v}^{\mathrm{d}}_k$ and $\tilde{v}^{\mathrm{d}}_{k+1}$ where the velocity forecast is derived directly from mean traffic speed, e.g. [107] or [108]. This reduces the forecast error because the acceleration necessary to reach $\tilde{v}^{\mathrm{d}}_{k+1}$ and thus the acceleration force is included. Using the derived acceleration profile to extend the standard model presented above is denoted $\tilde{E}^{\mathrm{d}}_{c|a=const}$. This approach represents the intuitive solution and usually, it is assumed that the resulting error is small enough without explicitly validating it with test drive data, e.g.:

> Since the length of each route segments from Google Map is usually quite small, using constant acceleration for each route segment based on the capability of the EV and driver's behaviour as well as the speed limitation produces negligible error. [107, p. 2]

Other researchers implement a driver model which learns situational acceleration behaviour, for example when approaching a traffic light or changed speed limit [45, 65]. These approaches rely on infrastructure data such as traffic lights state which are nowadays not available on navigation systems. Arguably, it will also not be available in the future since many traffic lights are controlled based on short-term demand (within seconds) instead of a fixed timetable [90]. Consequently, such models may yield good performance but are based on a lot of input data not accessible at the route planning stage, a fact supported by [69]. Additionally, situational acceleration behaviour is unlikely to be predictable with high accuracy due to local traffic and driver behaviour. Intuitively, even if the stop signs are in the TRDB data, it will never be known whether another vehicle will force the driver to stop there when he arrives at a specific location one hour in the future. Therefore, such an approach is not followed in this dissertation, instead a model is created based on the currently available TRDB data.
An alternative to inducing constant acceleration, the kinetic energy change between the segments k and $k+1$ can be included. This model, denoted $\tilde{E}_c^{\mathrm{d}}$, is based on the standard model $\tilde{E}_{c|a=0}^{\mathrm{d}}$. Thereby, $\tilde{E}_{c|a=0}^{\mathrm{d}}$ includes the static characteristics of the BEV energy consumption, e.g. potential energy change. The powertrain losses η_{stat} implicit with the constant power demand are obtained from the characteristic maps and also incorporated in the model $\tilde{E}_{c|a=0}^{\mathrm{d}}$ already. To extend it with the dynamic characteristics, the kinetic energy change is added which is also subjected to losses, represented

by η_{dyn}. While η_{stat} can be calculated from the model directly, η_{dyn} is an adaptive parameter minimizing the error $e_{E_{\text{c}}} = \tilde{E}_c^{\text{d}} - E_{\text{c}}$. By using this practical, data-driven approach with a linear factor η_{dyn}, over-fitting to a certain driver or trip is unlikely. (4.6) shows the prediction model for calculating $\tilde{E}_{c,k}^{\text{d}}$:

$$\tilde{E}_{c,k}^{\text{d}} = \underbrace{\int \frac{\tilde{F}_{\text{r},k}}{\eta_{\text{stat}}(\tilde{F}_{\text{r},k}, \tilde{v}_k^{\text{d}})} \text{d}s}_{\tilde{E}_{c|a=0}^{\text{d}}} + \frac{\hat{m}}{2\eta_{\text{dyn}}} \underbrace{[(\tilde{v}_{k+1}^{\text{d}})^2 - (\tilde{v}_k^{\text{d}})^2]}_{\Delta(\tilde{v}^{\text{d}})^2} \tag{4.6}$$

$$\eta_{\text{stat}} = f(\tilde{F}_{\text{r},k}, \tilde{v}_k^{\text{d}});\ \eta_{\text{dyn}} = f(e_{\tilde{E}_{c|a=0}}) \tag{4.7}$$

The mass estimation $\hat{m}$ in (4.6) is provided by the PE algorithm presented in Section 4.1.1. To verify the performance of the created model, it is compared to $\tilde{E}_{c|a=0}^{\text{d}}$ and $\tilde{E}_{c|a=const}^{\text{d}}$ in Section 4.1.3. Thereby, the implementation of the constant acceleration model from [108] is used, where the future acceleration $\tilde{a}_k^{\text{d}}$ on every segment k can be calculated using the time forecast $\tilde{t}_k^{\text{d}}$ on each segment comprising the segment length and vehicle speed forecast:

$$\tilde{a}_k^{\text{d}} = \frac{\tilde{v}_{k+1}^{\text{d}} - \tilde{v}_k^{\text{d}}}{\tilde{t}_k^{\text{d}}} = \frac{\tilde{v}_{k+1}^{\text{d}} - \tilde{v}_k^{\text{d}}}{l_k} \cdot \tilde{v}_k^{\text{d}}\,. \tag{4.8}$$

The energy consumption models presented in this section all result in deterministic predictions. After validating them in the following section, the methodology to achieve a stochastic solution will be shown in Chapter 5.

4.1.3 Performance Evaluation

In this section, the performance of the energy consumption model using PE from Section 4.1.1 and its predictive extension from Section 4.1.2 shall be examined. To validate and compare deterministic models, it is advantageous to use robust performance measures, therefore they are not blurred in the presence of a few outliers. This is similar to the problem of robustness for recursive estimators presented in Section 4.1.1. Consequently, the oftenly used mean value in performance measures should be avoided and the median used instead. For details on the issue of robustness in signal processing and measuring estimation performance, see [149]. This dissertation uses robust versions of the mean percentage error (MPE) and mean absolute error (MAE), called median percentage error (MEDPE) and median absolute error (MEDAE) following the example of [100]. They are introduced in the following equations, where the true value of a variable x is compared with the model estimation $\hat{x}$:

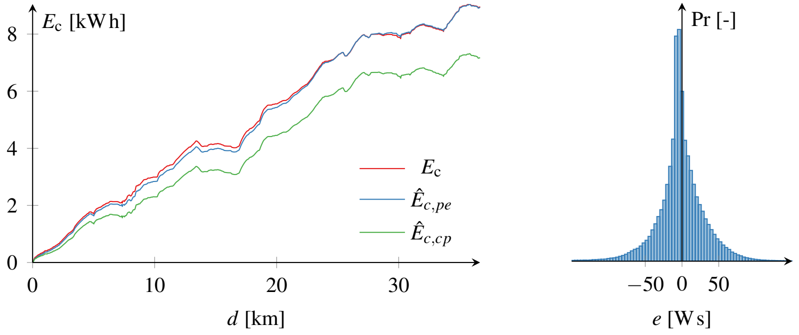
$$\text{MEDPE} = \underset{i}{\text{med}} \left(\frac{x_i - \hat{x}_i}{x_i} \right) \tag{4.9}$$

$$\text{MEDAE} = \underset{i}{\text{med}} \left(|x_i - \hat{x}_i| \right) \tag{4.10}$$

Using these performance indices, firstly the energy consumption model using PE is evaluated. For this, the driving resistances F_r are calculated with the parameter estimate $\hat{\mathbf{p}}$, and together with the powertrain model from Section 2.1, the energy consumption estimate $\hat{E}_{c,pe}$ can be calculated. This estimate for energy consumption can consequently be compared to the true value E_c obtained from measurements. Another possibility to estimate energy consumption is using constant test bench parameters instead of the PE results, which is a model frequently used in research, e.g. [107, 69]. Those constant parameters for example use the mass of the empty vehicle plus one driver, average air density for the location of the test drives etc. With these constant values and the powertrain losses, the energy consumption estimate $\hat{E}_{c,cp}$ can be calculated.

The results of the two different models on one exemplary test drive with the BEV can be seen in Figure 4.6a. From the course of the cumulated energy consumption it can be seen that the model with PE $\hat{E}_{c,pe}$ is significantly better than the model using standard parameters $\hat{E}_{c,cp}$, since it is closer to E_c.

Qualitatively, the course of $\hat{E}_{c,cp}$ follows is similar to E_c but they drift further away over the distance of the test drive. From the qualitative and quantitative profile, it seems that parameter estimation leads to an accurate energy consumption model, however this selected test drive could be a positive exception. Therefore, the total error histogram for $\hat{E}_{c,pe}$ on all 380 km driven with the BEV is shown in Figure 4.6b to provide a complete analysis. The expected value of the error distribution is almost zero therefore no systematic model error is present and the exemplary test drive from Figure 4.6a is no exception in the total test drive dataset. The remaining uncertainty originates from the random error signified by the variance of the distribution.



a) Cumulated energy consumption

b) Total error histogram

Figure 4.6: Performance of the energy consumption models. Figure 4.6a shows the comparison of the true energy consumption E_c with the model using parameter estimation $\hat{E}_{c,pe}$ and the model using constant parameters $\hat{E}_{c,cp}$ for one exemplary test drive of 37 km. Figure 4.6b illustrates the total error histogram on all 380 km of the BEV test data set for the superior model, $\hat{E}_{c,pe}$.

Apart from the visual performance evaluation based on Figure 4.6, the performance measures presented above, MEDPE and MEDAE, can be used to quantify the model accuracy. Table 4.1 shows the results for the model using parameter estimation (PE) and using constant parameters (CP) for both the exemplary test drive of 37 km (Figure 4.6a) and the total data set of 380 km.

From this table, it can be confirmed that $\hat{E}_{c,pe}$ with an overall MEDPE only $-2.1\,\%$ is a significant improvement compared to $\hat{E}_{c,cp}$ with $14.4\,\%$. While the model is not perfect, e.g. it does not include disturbance forces such as wind speed, it is valid to develop the range estimation method depending on its energy consumption characteristics. It could be shown by this evaluation that including PE poses a substantial improvement compared to the standard approach in literature, which uses constant parameters.

Table 4.1: Error of the energy consumption models, quantified using the robust performance measures.

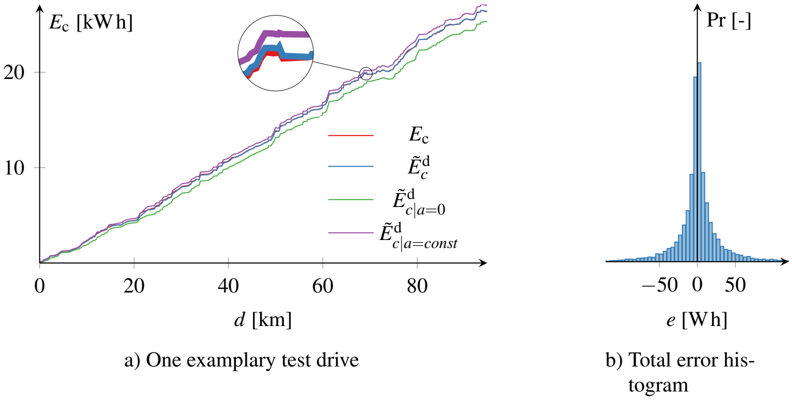
	MEDPE	MEDAE
PE (37 km)	1.2 %	17.3 W s
PE (380 km)	−2.1 %	14.4 W s
CP (37 km)	15.7 %	29.6 W s
CP (380 km)	14.4 %	27.4 W s

After validating the energy consumption model with PE in general, the predictive extension presented in Section 4.1.2 must also be analysed. This analysis is executed in the same way as for the $\hat{E}_{c,pe}$ model above. Since the energy consumption model has been validated already, now the test drive data of the ICEV can be used, which is bigger and therefore offers more statistical validity. For $\tilde{v}^{\mathrm{d}}$, the true mean from the test drive is used which represents the ideal deterministic prediction. In a forecast application, this ideal forecast is replaced by the velocity prediction (VP) explained in Section 5.1.

As introduced in Section 4.1.2, the standard model $\tilde{E}^{\mathrm{d}}_{c|a=0}$ and the constant acceleration model $\tilde{E}^{\mathrm{d}}_{c|a=const}$ are compared with the kinetic energy approach $\tilde{E}^{\mathrm{d}}_{c}$. Figure 4.7a shows the result for one exemplary test drive of 95 km. It can be seen that while $\tilde{E}^{\mathrm{d}}_{c|a=0}$ follows the qualitative course of E_{c}, it drifts away from the true energy consumption. This supports the statement that using the energy consumption model without extending it with a measure of acceleration could result in predictive bias. Since in such a model, part of the resistance force is omitted, the energy consumption will be under-estimated and in a range estimation application, this could result in unpredicted stranding. The constant acceleration model $\tilde{E}^{\mathrm{d}}_{c|a=0}$ is qualitatively closer to E_{c}, showing this model extension reduces the predictive bias already. The best performing model is the kinetic energy approach $\tilde{E}^{\mathrm{d}}_{c}$, however in Figure 4.7a it is almost exactly on E_{c} which is why it is visualized in the magnification. To evaluate the error of $\tilde{E}^{\mathrm{d}}_{c}$ over all the test drives on a total of 1900 km, the histogram Figure 4.7b is illustrated. The mean value of the error distribution $\mu = 0$ indicates the absence of a systematic error and the standard deviation is $\sigma = 37.25\,\mathrm{Wh}$. Compared to the error histogram of the estimated energy consumption in Figure 4.6b, it stands out that the absolute error values are higher because in the prediction model, the energy is calculated on the whole segment rather than per time step. In fact, the segments on highway drives are often longer than 1 km while the driven distance on a 10ms timestep and thus the energy consumption is significantly smaller.

To get a quantitative evaluation, the above presented robust error measures MEDPE and MEDAE are also applied to the prediction models and shown in Table 4.2. It can be seen that both on the exemplary test drive as well as the total dataset, the kinetic model $\tilde{E}^{\mathrm{d}}_{c}$ yields the best results, closely followed by the constant acceleration model $\tilde{E}^{\mathrm{d}}_{c|a=const}$. The standard model without extension shows large errors with MEDPE 13.8 % on all test drives. Both other models are accurate enough for a predictive application with total MEDPE values below 2 %. In the following, model $\tilde{E}^{\mathrm{d}}_{c}$ is used for the energy consumption forecast since it is the best overall alternative.

In summary, Section 4.1 showed the complete energy consumption model using both microscopic models and data-driven algorithms. It was shown that parameter estimation (PE) significantly improves model quality leading to an accurate simulation result, whereas the frequently used approach of using constant driving resistance parameters results in a biased solution. Furthermore, the extension for predictive applications is also required to



a) One examplary test drive

b) Total error histogram

Figure 4.7: Performance of the prediction models. Figure 4.7a shows the comparison of the standard prediction model without the extension $\tilde{E}_{c|a=0}^\mathrm{d}$ with the kinetic energy model $\tilde{E}_c^\mathrm{d}$ and the constant acceleration model $\tilde{E}_{c|a=const}^\mathrm{d}$ for one test drive of 95 km. In the magnification, the difference between $\tilde{E}_c^\mathrm{d}$ and E_c can be seen. Figure 4.7b illustrates the total error histogram on all 1900 km of the ICEV test data set for the best model, $\tilde{E}_c^\mathrm{d}$.

Table 4.2: Error of the energy prediction models, quantified using the robust performance measures.

	MEDPE	MEDAE
$\tilde{E}_c^\mathrm{d}$ (95 km)	2.5 %	8.9 Wh
$\tilde{E}_c^\mathrm{d}$ (1900 km)	1.6 %	8.7 Wh
$\tilde{E}_{c|a=const}^\mathrm{d}$ (95 km)	2.6 %	8.7 Wh
$\tilde{E}_{c|a=const}^\mathrm{d}$ (1900 km)	1.8 %	8.8 Wh
$\tilde{E}_{c|a=0}^\mathrm{d}$ (95 km)	12.1 %	11.9 Wh
$\tilde{E}_{c|a=0}^\mathrm{d}$ (1900 km)	13.8 %	11.8 Wh

generate suitable forecasts. In the following of this dissertation, the presented energy consumption model will be used to generate deterministic as well as stochastic forecasts which will be explained in Chapter 5.

4.2 Traffic Phase Classification

In the previous section, the energy consumption model was introduced which can be used to generate forecasts $\tilde{E}_\mathrm{c}$ based on the estimated driving resistance parameters $\hat{\mathbf{p}}$ and the velocity prediction $\tilde{v}$. Thereby, the parameter estimation (PE) algorithm was already presented in Section 4.1.1 which relies on physical equations and recursive algorithms. The other important factor defining energy consumption, driving speed v, strongly depends on driving behaviour and traffic situation, therefore the explanatory power of physical equations is limited. Hence, in this work driving behaviour is learned with a data-driven approach using features from the TRDB (Section 5.1). However, since driving behaviour strongly depends on the surrounding traffic situation, it is advantageous to provide an accurate local estimate.

The traffic situation can be determined using the traffic speed u, flow f_u and density ρ_u as well as the fundamental diagram of traffic flow [38]. Thereby, the overall traffic situation can be divided into three macroscopic traffic phases following Kerner's theory [67]: free-flow (FF), synchronized-flow (SY) and wide moving jam (J). These traffic phases can be identified with help of the fundamental diagram and probe data, which is acquired traditionally from loop detectors in the road [66]. In recent years, the number of connected vehicles increased consistently, therefore this source of information can also be used to identify the traffic phase [68, 134]. The motivation for the transportation research community in this area is usually the correct prediction of travel times and hence estimated time of arrival (ETA) for navigation purposes [94]. Thus, the type of traffic situation information is averaged on road segments and lanes. The TRDB data used in this dissertation also follows this principle.

Learning situational driving behaviour however requires the local traffic phase in the direct surrounding of the ego-vehicle instead of an averaged value. Consequently a new algorithm has to be implemented that uses internal measurement data in real-time. Using that local traffic phase estimate furthermore avoids errors dependent on e.g. time delay resulting from the communication of road sensors and probe vehicles with the traffic database or limited number of connected vehicles. In addition, if many connected vehicles use local traffic phase estimation, the global traffic phase identification would be significantly easier for traffic providers. The following section presents the details of the traffic phase classification (TPC) algorithm for the individual ego-vehicle which has already been published to the research community in [111].

4.2.1 Feature Selection

To identify the local traffic phase ϕ in the near surrounding of the ego-vehicle, various sources of vehicle sensors are available, e.g. camera or radar information. Among those sensor signals, many provide an indication of the traffic phase but no signal explicitly identifies it. For example, if the vehicle speed v is very close or even above the speed limit v_{lim}, it is likely that the local traffic phase ϕ is free-flow (FF). However, if v is lower than v_{lim}, the traffic phase ϕ cannot be inferred because it might be the drivers choice to be slower than legally allowed. Another example for a relevant feature would be the detection of a vehicle ahead of the ego-vehicle $\mathrm{o}^{(1)}$, which is a binary signal. Radar and camera cannot only identify whether a vehicle is ahead, but up to 5 vehicles on three lanes, as illustrated in Figure 4.8. In wide moving jam (J) and synchronized-flow (SY), it is very likely that a leading vehicle $\mathrm{o}^{(1)}$ is detected and less likely in free-flow (FF). Using the test drive data from the ICEV introduced in Section 3.4, the characteristics of the two features mentioned above can be shown. Thereby, the traffic phase was manually labelled in the dataset.

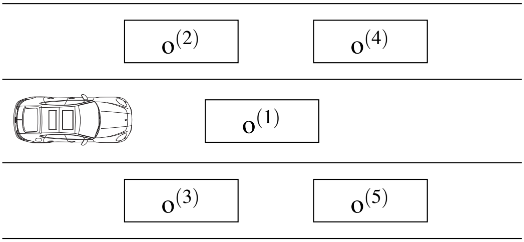


Figure 4.8: Orientation and denotation of the vehicles detected by camera and radar of the ICEV. Vehicles 4 and 5 can only be identified if 2 and 3 are visible, respectively.

Figure 4.9a shows that in J, the ratio of driven speed and speed limit v / v_{lim} is significantly lower than in SY and FF, as expected. Thereby, v / v_{lim} is a continuous feature, hence illustrated as box-plot. The other considered feature o$^{(1)}$ is discrete, therefore the empirical probability Pr(o$^{(1)}$) is shown in Figure 4.9b. It can be seen that Pr(o$^{(1)}$) is much lower in FF than in SY and J which was also expected. From this example, it is apparent that none of the two features is able to properly separate all three phases. With v / v_{lim}, SY and FF cannot be distinguished properly whereas with Pr(o$^{(1)}$), SY and J are not distinguishable. However, it seems that a combination of the two features might result in a good estimate for ϕ. To properly combine a set of features is the task of the classification algorithm presented in Section 4.2.2. Before designing that algorithm, the relevant features must be extracted from the available signals, which is called *feature selection*. Fundamentals of feature selection methods can be found in [43].

The central task of feature selection is to analyse the individual relevance of the available signals and offer a numeric quantity to compare them. For continuous features, this can be achieved with the analysis of variance (ANOVA). In ANOVA, the observations in several classes are compared to determine whether it is likely that the underlying process is one

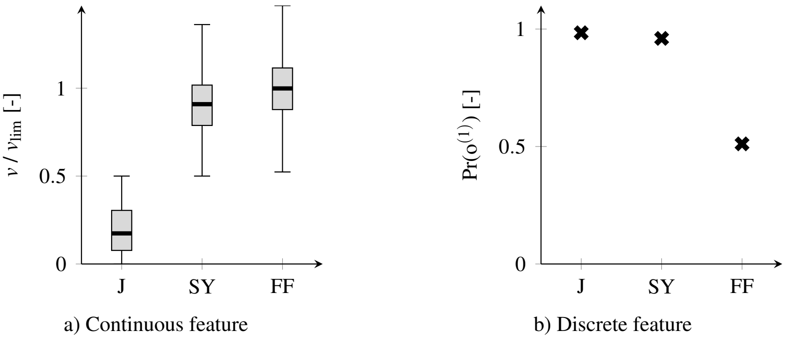


Figure 4.9: Two exemplary features for traffic phase identification with data from the labelled ICEV dataset. In Figure 4.9a, a continuous feature is shown which is able to distinguish between the traffic phase J and (SY,FF). Figure 4.9b illustrates a discrete feature which separates the traffic phase FF and (J,SY).

singular distribution [37]. If that *null hypothesis* is true, the mean values of the observations in the individual classes would be the similar. However, even if the mean values are different, a very large variance would also make the classes not separable. In a way, it can be considered as the mathematical equivalent of comparing the distributions for each class with the boxplots in Figure 4.9a. Therefore, the relevance of a signal using ANOVA is calculated with help of the means and variances of the observations following (4.11), called the F-test [78]:

$$F = \frac{\frac{1}{I-1}\sum_{i=1}^{I} N_i(\bar{o}_i - \bar{o})^2}{\frac{1}{N-I}\sum_{i=1}^{I}\sum_{j=1}^{N_i}(o_{ij} - \bar{o}_i)^2} \tag{4.11}$$

where N is the total number of observations, I is the number of classes, o_{ij} is the individual observation, $\bar{o}_i$ the mean value of observations in class i, $\bar{o}$ the overall mean of all observations, N_i the number of observations in class i and F the resulting relevance quantity. For an examplary implementation of ANOVA, see [116].

After calculating F for the candidate signals of the CAN, they can be ranked according to their relevance which is shown in Table 4.3. In addition to the F result, the correlation with other features can be seen. If the correlation coefficient with another variable is above 0.5, then the variable is listed in the last column. Furthermore, the separated classes are indicated derived from boxplot visualizations equivalent to the example in Figure 4.9a. The listed features apart from $v\,/\,v_{\text{lim}}$ are the distance to the vehicle ahead $d^{(1)}$, the relative speed to the other detected vehicles $v_{\text{rel}}^{(1-5)}$, the time headway to the vehicle ahead $\text{thw}^{(1)}$ and the acceleration of the ego-vehicle a. From the results, it can be seen that the feature $v\,/\,v_{\text{lim}}$ is the most relevant variable, followed by the information about the vehicle ahead in terms of distance, relative speed and time headway. In contrast, information about the other surrounding vehicles is less important and from the visual analysis, the signals do not help separate traffic phases.

Table 4.3: Ranking of continuous features with the ANOVA method. Apart from the F-result, the separated classes as well as the correlated variables with a coefficient of > 0.5 are shown.

feature	F	separated classes	correlation
$v\,/\,v_{\text{lim}}$	$1.3 \cdot 10^6$	J - (SY, FF)	$d^{(1)}$
$d^{(1)}$	$3.0 \cdot 10^5$	J - SY - FF	$v\,/\,v_{\text{lim}}$
$v_{\text{rel}}^{(1)}$	$1.7 \cdot 10^5$	(J, SY) - FF	-
$\text{thw}^{(1)}$	$4.9 \cdot 10^4$	(J, SY, FF)	-
$v_{\text{rel}}^{(5)}$	$1.7 \cdot 10^4$	(J, SY, FF)	$v_{\text{rel}}^{(3)}, v_{\text{rel}}^{(4)}$
$v_{\text{rel}}^{(2)}$	$1.6 \cdot 10^4$	(J, SY, FF)	-
$v_{\text{rel}}^{(3)}$	$1.1 \cdot 10^4$	(J, SY, FF)	$v_{\text{rel}}^{(4)}, v_{\text{rel}}^{(5)}$
$v_{\text{rel}}^{(4)}$	$6.9 \cdot 10^3$	(J, SY, FF)	$v_{\text{rel}}^{(3)}, v_{\text{rel}}^{(5)}$
a	$7.8 \cdot 10^2$	(J, SY, FF)	-

After analysing the continuous features, the relevance of the discrete signals must be evaluated. This can be achieved with the χ^2 test developed by [92]. The χ^2 performs a similar evaluation compared to ANOVA, as it tests against the null hypothesis of independence between the observations and the classes. Applied on the problem at hand this would transform to e.g. whether the probability of detecting a leading vehicle $\mathrm{o}^{(1)}$ is independent of the traffic phases. The hypothesis can be analysed with the formula for χ^2-statistics:

$$\mathrm{X}^2 = \sum_{i=1}^{I} \frac{(O_i - H_i)^2}{H_i} \tag{4.12}$$

where the X^2 value is calculated based on the total number of classes I, observed frequency O in the class i and the expected frequency from the null hypothesis H in the classes i. Using the example of Figure 4.9b, the relative frequency $\frac{O}{N} \approx 0.5$ was observed in class FF. Given the null hypothesis, it would be expected that there is an equal number of observations in the classes, hence $\frac{O}{N} \approx \frac{1}{3}$. Similar to the F-test, the resulting X^2 value of (4.12) can be interpreted as how relevant the signal is and therefore, it can be used to compare the features. The χ^2 test is applied on the candidate signals from the CAN-bus and the most relevant discrete features are shown in Table 4.4. Those features comprise the detected objects $\mathrm{o}^{(1-5)}$ as well as the identification of a completely free lane FL with

$$\mathrm{FL} = \neg\mathrm{o}^{(1)} \vee (\neg\mathrm{o}^{(2)} \wedge \neg\mathrm{o}^{(4)}) \vee (\neg\mathrm{o}^{(3)} \wedge \neg\mathrm{o}^{(5)}) \,. \tag{4.13}$$

Apart from the individual χ^2 test result, the separated traffic phases are shown after visual analysis following the example of Figure 4.9b. In addition, the correlation with other features can be seen, equivalent to Table 4.3. From the results, it seems that the discrete features are primarily suitable to identify free-flow (FF). Furthermore, signals $\mathrm{o}^{(2-5)}$ are correlated with each other. The most important features are the detection of a vehicle ahead $\mathrm{o}^{(1)}$

and identification of a free lane FL. Among the other vehicles detected, the highest relevance is achieved if two vehicles are identified on the right lane.

Table 4.4: Ranking of discrete features following the score of the χ^2 test. Additionally, the information which traffic phases can be classified based on a specific feature is shown. The last column lists other features with a correlation coefficient of above 50 %.

feature	X^2	separated classes	correlation
$o^{(1)}$	$2.5 \cdot 10^5$	(J, SY) - FF	-
FL	$6.0 \cdot 10^4$	(J, SY) - FF	-
$o^{(5)}$	$3.3 \cdot 10^4$	(J, SY) - FF	$o^{(3)}$, $o^{(4)}$
$o^{(2)}$	$3.3 \cdot 10^4$	J - SY - FF	$o^{(4)}$
$o^{(3)}$	$2.9 \cdot 10^4$	(J, SY, FF)	$o^{(5)}$
$o^{(4)}$	$2.2 \cdot 10^4$	(J, SY, FF)	$o^{(2)}$, $o^{(5)}$

4.2.2 Classification Algorithms

The mathematical problem of identifying the correct traffic phase among a finite set of possibilities is called *classification*. Classification in general and specific suitable algorithms for different kinds of problem formulations have been studied by other researchers, see [47] for an overview. Among all the candidate algorithms, in this work two approaches have been studied: rule based methods and graphical models. The former comprises heuristic algorithms with low computational or modelling complexity. They are based on a given set of decision rules created by the user, typically in an *if-then* structure. Rule-based control algorithms are known to automotive engineers from a variety of HEV operation strategies [105]. Thereby, the rules can be evaluated deterministically with classical logic or using fuzzy logic. Classical logic is of a binary nature, a statement can be *true* or *false*. In fuzzy logic, a statement can be true *to some degree*, thus instead of a binary result, any real number between 0 and 1 can be assumed indicating the uncertainty. For a general overview about logical approaches, see [138]. To implement the rule-based approaches, first the rule set must be determined which is created by expert knowledge. Depending on the type of logic, those rules are evaluated by both the standard rule-based algorithm (RBA) and fuzzy-logic algorithm (FLA). The rules which can clearly identify the traffic phase are called *explicit*. For the classical logic approach in RBA, an additional *implicit* rule set can be used if no explicit rule is applicable or if there is a conflict. The complete rule set is illustrated in Table 4.5.

Table 4.5: Rule set for traffic phase classification with explicit rules defining ϕ and implicit rules giving an indication. The rules were generated using expert knowledge. The individual bullet points within a phase are connected by a logical $\vee$

ϕ	explicit	implicit
FF	• No leading vehicle detected • Free lane in sight • driven speed 20 km/h above speed limit	• fast overtaking • leading vehicle drives faster $\wedge$ far away
SY	• leading vehicle detected $\neg$ low speed $\neg$ very high speed $\neg$ significant stoppage	• speed relatively constant • time headway relatively constant
J	• leading vehicle detected as well as other vehicles $\wedge$ low speed • vehicle stop, not at intersection	• small headway • vehicles detected on all lanes • frequent acceleration and deceleration

Apart from the rule-based methods, data-driven graphical models using Bayesian statistics are applied which represent a mathematically more complex approach to the problem of traffic phase classification (TPC). The general model structure used in this dissertation is called a Bayesian net (BN). BNs comprise a set of variables A_i as nodes and their conditional dependencies as edges using a *directed graph* following [35]. The network is defined with

$$\Pr(A_1,...,A_n) = \prod_{i=1}^{n} \Pr(A_i|\Pi_{A_i}) = \prod_{i=1}^{n} \theta_{A_i|\Pi_{A_i}} \tag{4.14}$$

where Π_{A_i} denotes the parents of node A_i signified by the model edges and $\Pr(A_i|\Pi_{A_i})$ denotes the conditional probability. It can be seen that the BN depends on both the graph structure which is built by the user and the conditional probabilities between connected nodes, known as BN parameters θ. Without prior knowledge about the dependencies, these parameters must be learned from data. For that, the standard *maximum likelihood method* is used, see [81]. Thereby, discrete features are represented as multinomial distributions and continuous features with normal distributions. Those are created with the empirical probabilities from the training data as in Figure 4.9. For example, assuming a simple model structure where the observation variable $\mathrm{o}^{(1)}$ has only one parent ϕ, the parameters θ can directly be obtained from the empirical probabilities shown in Figure 4.9b, e.g.:

$$\theta_{\mathrm{o}^{(1)}=1|\phi=\mathrm{FF}} = \frac{O_{(\mathrm{o}^{(1)}=1)}}{N_{(\phi=\mathrm{FF})}} \tag{4.15}$$

where $\theta_{\mathrm{o}^{(1)}=1|\phi=\mathrm{FF}}$ is the conditional probability of FF given a detected vehicle ahead $\mathrm{o}^{(1)} = 1$, the number of detections in the specific traffic phase is $O_{(\mathrm{o}^{(1)}=1)}$ and the total number of observations in FF is $N_{(\phi=\mathrm{FF})}$. With the defined model structure and the learned dependencies, a BN can be used to infer the *hidden* state of a variable given a new set of observations. The hidden variable in the TPC algorithm is the traffic phase ϕ which cannot be measured, but instead must be inferred given the selected features. In the following, the specific model structures for TPC are introduced, since the BN parametrization follows standard procedures, see [20] for a complete overview.

Three types of BNs were implemented, each algorithm with a different observation structure. The simplest case of observation structure would be to assume all features are independent of each other. Using this type of BN in a classification algorithm is called naive Bayesian classifier (NBC). The resulting model structure and the selected features are illustrated in Figure 4.10. It can be seen that the nodes indicating the features are not connected with each other and only with the class variable ϕ because of the independence assumption. The best three continuous features and the best four discrete features were selected for the NBC. Despite its simplicity, NBC shows good performance in various applications, e.g. [145].

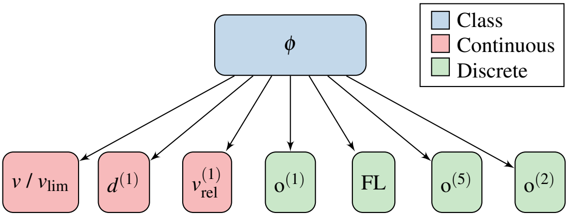


Figure 4.10: Model structure of a NBC which represents a BN with independent features. In consequence, the edges only connect the class variable with the individual features but no edges connect the features with each other.

The observation structure of the NBC can be extended by including the dependencies between features, e.g. the relative velocity of the leading vehicle $v_{\text{rel}}^{(1)}$ can only be observed if a leading vehicle is detected $\text{o}^{(1)}$. Also, a vehicle to the left $\text{o}^{(2)}$ can only be observed if the ego-vehicle is not on the left lane, a variable depicted LL. Equivalently, the information whether the ego-vehicle is on the right line is depicted RL. The information also influences the calculation of FL following (4.13). Including this extension turns the NBC into a classical Bayesian net (BN), illustrated in Figure 4.11. There, the model extension compared to the NBC is apparent by the extra edges connecting features among each other. In addition to the ability to consider the inter-dependencies, BNs were shown to be robust to label noise [32] which is an important characteristic given manually labelled training data.

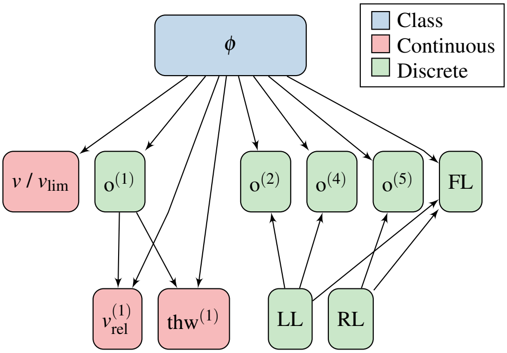


Figure 4.11: Model structure of a BN which considers the inter-dependencies of the features. Therefore the edges not only connect the features with the class but also the features among each other.

A further extension would be the consideration of time dependency in TPC. Such a model could improve classification performance since traffic flow is a spatio-temporal phenomenon. For example, it is unlikely that ϕ changes directly from free-flow (FF) to wide moving jam (J) without being in synchronized-flow (SY) before. It is also unlikely that the traffic phase changes in every single time-step at a high frequency. Including these so called *transition probabilities* with respect to time requires a hidden Markov model (HMM), which is a special case of BN according to [81]. HMMs require an observation structure and the conditional dependencies θ in the same way as a BN. In addition, the class variable ϕ depends on its previous state following the commonly known Markov assumption [89]. Thus, a transition probability T_{ϕ_1,ϕ_2} is assigned to all possible combinations, as illustrated in Figure 4.12. The quantitative values for T_{ϕ_1,ϕ_2} are also learned from empirical transition frequencies within the training data. For a general introduction into HMMs, see [95]. In a similar mathematical problem, the HMM was shown to achieve good performance in classifying discrete driving states during real test drives [36], hence it is also expected to perform well in the TPC application. In the following, the HMM with a BN observation structure is called dynamic Bayesian net (DBN) and the HMM with a NBC observation structure is called dynamic naive Bayesian classifier (DNBC).

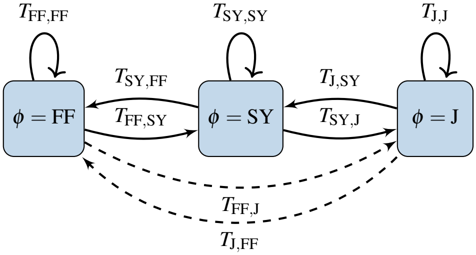


Figure 4.12: HMM model structure showing the transition probabilities T_{ϕ_1,ϕ_2} indicating the time-dependency of traffic phase on its previous state. The direct transition between FF and J is highly unlikely, indicated by the dashed edges.

In summary, six different classification algorithms were developed to identify ϕ. In the order of increasing model complexity, they are the rule-based algorithm (RBA), fuzzy-logic algorithm (FLA), naive Bayesian classifier (NBC), Bayesian net (BN), dynamic naive Bayesian classifier (DNBC) and the dynamic Bayesian net (DBN). The performance of these methods is evaluated in the following section.

4.2.3 Performance Evaluation

For evaluating the performance of classification algorithms, several possible error measures are available. In this work, the balanced error rate (BER) is chosen, a frequently used candidate which is defined as follows [96]:

$$\mathrm{BER} = \frac{1}{C}\sum_{i}\frac{\left(\sum_j M_{ij}\right) - M_{ii}}{\sum_j M_{ij}} \tag{4.16}$$

C is the number of classes and M is the $C \times C$ confusion matrix with M_{ij} being the number of times that traffic phase i was classified as phase j. For each algorithm, the classification results of all 10 test drivers was evaluated in a 10-fold cross validation scheme, as illustrated in Figure 4.13. That means, the data of one driver is reserved for testing and the data of all other drivers are used for training the algorithms in every fold. That has the side effect of ensuring the algorithms are independent of driving behaviour. The mechanism of cross-validation in general was shown in Section 2.4.

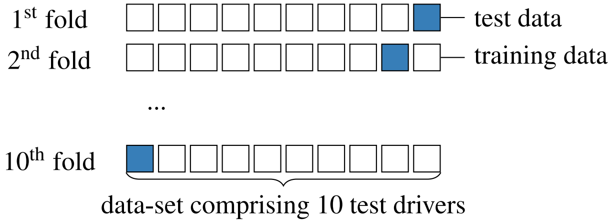


Figure 4.13: Cross-validation method for evaluating TPC algorithms. The complete dataset comprises data of 10 different test drivers. In a 10-fold cross validation, the data of one driver is selected as test data while the rest serves as training set.

The results from the cross-validation evaluated in terms of the BER are illustrated in Figure 4.14. They show that the graphical models (NBC, BN, DNBC, DBN) all perform better than the rule-based approaches (RBA, FLA). From that, the increased model complexity of inducing Bayesian statistics and training data seems to be worth the effort. However, the specific type of observation model does not influence the overall result significantly, it can even be seen that the NBC, the simplest algorithm among the graphical models, shows slightly better results than the others. This might seem puzzling as the assumption of independence within NBC is expected to turn out worse results, especially since there are apparent dependencies between the features. In fact, in various applications, the NBC proves to be a robust classifier despite the assumption of independence, and scientists research about the reason for it, e.g. in [122]. This application therefore provides another example of the NBC's potential, which is a simple and computationally efficient model that performs well. Looking at the results of the rule-based approaches, it can be seen that the normal logic method (RBA) shows adequate results, in the best case around 11 % error rate. This confirms that the defined rule set is suitable for the presented problem. The fuzzy-logic algorithm (FLA) performs worse because it does not consider the implicit rules. When alaysing the absolute values of BER, it must be kept in mind that 0 % is not necessarily the lower error bound because the algorithm classification is compared to manually labelled data. Naturally, data labelled by humans is subject to label noise. According to [34], it can be assumed to be around 5 %, which would then be the lower bound of BER. However, for a comparison of the algorithms among each other, this is of secondary importance since all have to deal with the same noise.

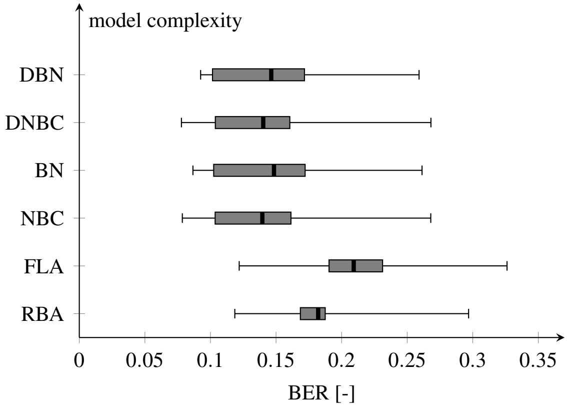


Figure 4.14: Boxplots of the BER results from the traffic phase classification algorithms for all test drivers. They are sorted by the modelling complexity of the inherent algorithms. The error values were evaluated using cross-validation on the total test drive set.

Table 4.6 illustrates the average BER of all test drives for the algorithms, hence the performance is reduced to one number. There, NBC is also the best algorithm with lowest $\overline{\text{BER}}$. In consequence, it is selected to provide information about the traffic phase to the velocity prediction (VP) algorithm presented in the following section. However, for an implementation in a modern day control box, it might not be feasible because training data is required and modelling complexity is increased compared to rule-based approaches. Therefore, the RBA results are also used to provide an estimate for ϕ to the VP in the following of this work. Thus, the importance of accurate TPC within the overall method can be analysed. The results are presented in Chapter 6.

Table 4.6: mean BER of the classification algorithms. The $\overline{\text{BER}}$ is gathered from the individual results shown in Figure 4.14 and the algorithms are sorted by their modelling complexity.

	RBA	FLA	NBC	BN	DNBC	DBN
$\overline{\text{BER}}$ [%]	18.561	21.440	14.717	15.204	14.723	15.047

5 Destination Attainability Forecast

This chapter introduces the algorithms necessary for creating the energy consumption forecast (ECF) and ultimately attainability following the proposed methodology in Section 3.2. Thereby, the approaches for obtaining estimates for the driving resistance parameters $\hat{\mathbf{p}}$ and the traffic phase $\hat{\phi}$ were already presented in Chapter 4, as indicated in Figure 5.1.

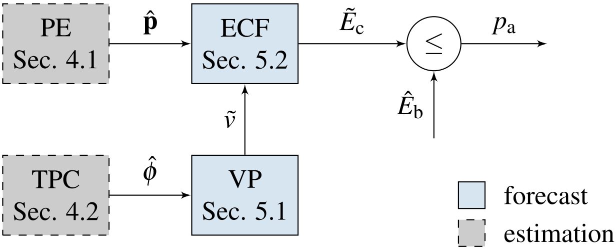


Figure 5.1: Overview of the approach for calculating attainability. Thereby, estimation algorithms PE and TPC were already presented in previous section and the required forecast algorithms are shown in this section: VP and ECF.

This chapter illustrates the forecast algorithms using the results of the mentioned estimation algorithms. In the following, Section 5.1 introduces the velocity prediction (VP) algorithm to obtain future velocity $\tilde{v}$ based on data-driven regression algorithms as well as TRDB information. The focus lies on explaining the used models, rather than evaluating the overall performance. After having explained the VP, the resulting $\tilde{v}$ can be used to generate the ECF, which is shown in Section 5.2. For stochastic forecasting, appropriate uncertainty propagation methods have to be used, which are also introduced in that section. Finally, obtaining attainability p_a with help of the future energy consumption $\tilde{E}_{\mathrm{c}}$ as well as the battery model is introduced in Section 5.3. Preliminary results of the complete method presented in this chapter of the dissertation were already introduced to the research community in [111].

5.1 Velocity Prediction

A vital influence on electric vehicle's energy consumption is the driving speed, as introduced in Section 2.1. Thus, for forecasting energy consumption following the method illustrated in Figure 5.1, a velocity prediction (VP) algorithm must be implemented, which is illustrated in this section. Velocity prediction algorithms for complete routes are known nowadays from navigation systems to obtain an estimated time of arrival (ETA). They rely on traffic information from probe data u and the resulting traffic speed prediction $\tilde{u}$, which is introduced in Section 5.1.1. Including driver behaviour in the forecast can be achieved by applying data-driven regression algorithms, which is demonstrated in Section 5.1.2. Thereby, two approaches are introduced: a simple model using a multi-linear regression (MLR) in Section 5.1.3 and a more complex machine-learnig model using a support vector machine (SVM) in Section 5.1.4. Both the traffic speed prediction and the regression-based prediction can be executed deterministically or stochastically, which is shown in the respective sections.

5.1.1 Ideal Traffic Speed Prediction

State-of-the-art velocity prediction algorithms for complete routes are mainly driven by the need to calculate accurate arrival times for navigation purposes. Commonly known solutions, e.g. HERE MAPS [48] or GOOGLE MAPS [41] use the mean traffic speed $\bar{u}$ of the connected probes to obtain the future velocity $\tilde{v}$ on every segment k. For the near future, directly using $\bar{u}$ for the prediction is adequate, but for a long-term horizon, the daily traffic pattern on the route has to be analysed because it cannot be assumed that the traffic situation remains unchanged. Instead, traffic providers try to extract the information of typical speed on the specific link at the specific time i.e. they learn the traffic pattern which changes during the day. Figure 5.2 provides an example for the mean traffic speed $\bar{u}$ on the segments k between Stuttgart and Heidelberg for every hour on the 13th of October 2017. It can be seen that depending on the time, mean traffic speed varies significantly over the course of the day. At some points, it beaks down to almost 20 % of the free-flow speed. Predicting accurate ETA strongly relies on the ability to find traffic patterns in the probe data and using them in the forecast, see [5, 101, 103] for research on this topic. It can be imagined that predicting velocity only based on the current traffic situation could lead to a biased result. For example, when predicting traffic speed on a segment several kilometres into the trip, the information about the *typical* situation on that segment on the time of arrival is more relevant than the speed of probes currently on that segment.

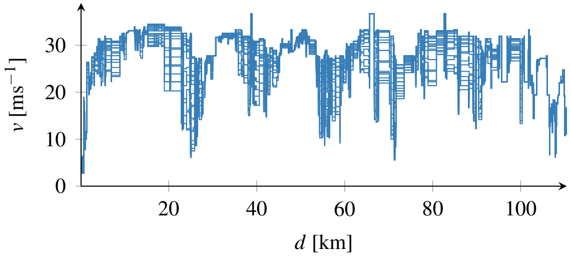


Figure 5.2: Mean traffic speed values on the route Stuttgart-Heidelberg for every hour on 13th October 2017.

In this work, the conventional velocity prediction using probe information shall be used for comparison with the proposed algorithms. Therfore, the problem of finding traffic patterns poses a challenge: If the traffic prediction of the traffic service provider is used, the prediction might show low performance because of undetected patterns or spontaneously emerging traffic jams on the route. Therefore, a method to acquire traffic information which is independent of the pattern analysis algorithm supplied by the provider must be used. Hence, instead of using the traffic prediction at the start of the trip, the current traffic information for specific time and position is downloaded during the test drive. This creates the *ideal* traffic prediction and represents the benchmark for the pattern analysis algorithms implemented by traffic providers. An excellent traffic forecast algorithm would be very close to the ideal traffic prediction. Therefore, it is also used as input for the predictive algorithms later in this work to exclude forecast errors of the traffic provider. Additionally, an error in the velocity forecast presented in this work reduces to errors produced by the model itself and not an inaccurate input variable. That is very important to ensure compatibility between different algorithms. Apparently, this methodology only holds for adequate data coverage. Since the test drives mostly took place on large highways, this is a valid assumption, also because [129] showed that around 40 % coverage sufficiently estimates traffic states and increasing coverage through connected vehicles can be expected in the future. At this point it must be said that in a real-world application, the input variable would be the current traffic speed information available at the beginning of the trip u_{current} instead of the ideal traffic speed u_{ideal}. The predictive quality depends on the provider but is presumably worse. Improving it is an important task for researchers and companies in this field.

A comparison of the ideal traffic speed u_{ideal} with the current traffic speed information available at the beginning of the trip u_{current} can be seen in Figure 5.3. In this example, traffic jams were on the route when the test vehicle began the trip, e.g. at around 80 km. During the time the vehicle needed to reach the specific segments, the traffic jam at 80 km had dissolved, as indicated by the higher u_{ideal} values. To prevent bias of this sort in the following, u represents the ideal traffic state. A prediction based on the ideal mean traffic speed $\bar{u}$ is denoted $\tilde{u}^{\text{d}}$, indicating a deterministic forecast.

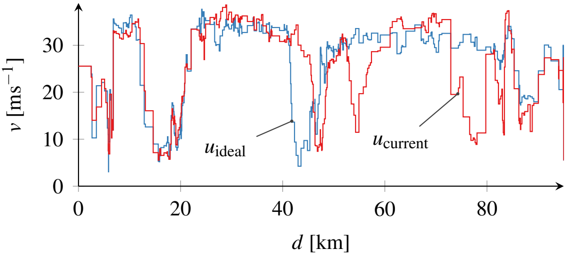


Figure 5.3: Comparison of the ideal traffic speed u_{ideal} and the current traffic speed at the beginning of the trip u_{current} for one exemplary test drive among the ICEV dataset.

As introduced in Section 2.3, a deterministic forecast neglects system uncertainties. However, even the ideal $\bar{u}$ is not a 100 % accurate velocity prediction for the individual vehicle because firstly, it does not include driving behaviour and secondly, it is based on a mean value and not all vehicles drive exactly according to the mean value. To illustrate how much $\bar{u}$ deviates from the driven speed v, one exemplary test drive of the ICEV dataset is used. The relative error of the ideal traffic speed compared to the averaged driven speed on every segment $e_u = (\bar{v} - \bar{u}) \cdot (\bar{u})^{-1}$ is calculated and shown in the histogram of Figure 5.4.

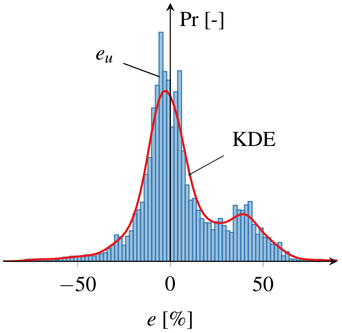


Figure 5.4: Example for the error distribution e_u of the deterministic ideal traffic speed prediction $\tilde{u}^{\text{d}}$. To illustrate the bi-modality of the distribution, the kernel density estimate (KDE) with a Gaussian kernel is included.

A significant amount of uncertainty can be seen with the 1st percentile being $-53\,\%$ and the 99th percentile $71\,\%$. In addition, the distribution does not seem to be uni-modal. This is illustrated with help of a kernel density estimation (KDE), signified with the red curve. One mode, the maximum a posteriori (MAP) estimate, is close to zero error which means the driving speed would be very close to the mean traffic speed. The other mode is close to $40\,\%$ error, which means there is a significant chance of the vehicle driving faster than the surrounding traffic. This bi-modality can be explained with Kerner's three phase theory [66], first introduced in Section 4.2. In the congested traffic phases (jammed and synchronized flow), the driver has little choice over the ego-vehicle speed v, therefore it is close to the mean traffic speed. In free-flow, the driver is not affected by other vehicles and can choose the driving speed freely, in this case significantly faster than surrounding traffic. From this example, it can be seen that predicting future velocity deterministically could lead to significant bias in the forecast. Even in the congested traffic phases with less systematic error, notable random error remains which is apparent in the uncertainty distribution. From these findings, it is expected that including the traffic phase estimate from Section 4.2 in the velocity prediction will lead to better forecast performance. However, this is evaluated in Chapter 6.

To achieve a stochastic velocity forecast, all the available probe data on a segment k can be used from the TRDB. Since all probes report their position and speed data, not only the mean value but also the total speed distribution of connected vehicles can be accessed. Using these data per segment creates the stochastic traffic forecast $\tilde{u}^{\mathrm{s}}$. Predicting speed in such a way uses the underlying assumption that the future vehicle speed will be within the boundaries of the probe distribution, hence an element of uncertainty is induced. Thus, the predictive distribution is equal to the ideal probe speed distribution. Figure 5.5 shows an exemplary prediction from the ICEV dataset using both the deterministic traffic prediction $\tilde{u}^{\mathrm{d}}$ and the stochastic forecast using probe information $\tilde{u}^{\mathrm{s}}$. Thereby, the $90\,\%$ prediction interval (PI) of $\tilde{u}^{\mathrm{s}}$

is indicated with the shaded area. The driven speed $\bar{v}$ is also included for comparison.

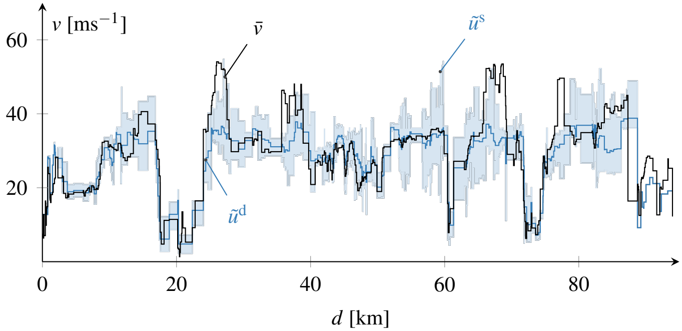


Figure 5.5: Exemplary prediction using ideal traffic information from the TRDB. $\tilde{u}^{\mathrm{d}}$ shows the mean traffic speed prediction conventionally used in modern navigation systems whereas $\tilde{u}^{\mathrm{s}}$ uses the complete probe distribution. Thereby, $\tilde{u}^{\mathrm{s}}$ is shown as the 90 % PI with the shaded areas. For comparison, the measured velocity $\bar{v}$ is included.

Again, a significant deviation of real driven speed v compared to deterministic forecast $\tilde{u}^{\mathrm{d}}$ can be seen, especially in some areas such as around 30 km or 70 km where the driving speed is notably above the prediction. Thereby, under-estimating future driving speed is especially disadvantageous in a range estimation application as it leads to under-estimation of energy consumption and ultimately an over-estimation of range. This could result in unpredicted stranding. Using the whole probe distribution in the stochastic forecast $\tilde{u}^{\mathrm{s}}$, the possibility of the driver choosing a velocity above $\tilde{u}^{\mathrm{d}}$ is included and except in some cases, v is within $\tilde{u}^{\mathrm{s}}$. From this example, the stochastic prediction seems to transfer more useful and more accurate information and is thus superior to the deterministic prediction. For a full analysis on the total test drive dataset, the reader is referred to Chapter 6.

5.1.2 Velocity Prediction using Regression Algorithms

Commonly, velocity prediction algorithms for complete trips are based on traffic information as shown in the previous section. If the algorithm relies only on probe information, the driver behaviour cannot be included. However, learning driving behaviour using data-driven regression algorithms could improve forecast performance. Predicting future velocity including driver behaviour can fundamentally be executed based on two forecast principles: time-series based forecasting and feature-based forecasting (see Section 2.3). The former uses the pattern within the past values of the velocity signal to predict its future profile, whereas the latter can model the dependency of explanatory variables (features) on the driven velocity. A commonly used time-series model is the Markov chain (MC) formulation, where the prediction is only dependent on the observation of the last time step. This model type has been used for vehicle speed prediction in literature, e.g. [24]. It shows good performance in short-term predictions, for example if the driving speed in the next seconds is of interest [57]. However, even on arelatively shorty prediction horizon of $\geq 10\,\mathrm{s}$, it was found that explanatory models give better forecast results, see [36]. In a long-term prediction application, this is even more likely. The driving behaviour in the last time step is less important than environmental factors such as traffic speed, street type or number of lanes when looking several hours into the future.

Explanatory forecasting is based on input variables, the features, and regression or machine learning algorithms, which extract the connection between the features and the outcome. Several approaches from linear models to deep learning have been applied in literature, an exemplary comparison of different algorithms can be seen in [71]. Still, the maximum prediction horizon considered in that publication was 10 s, which is too short for a range estimation application for long-distance trips. Forecasting vehicle speed up to 30 min into the future can be found in [91], where a neural network (NN) is implemented. Still, the approach is based purely on probe information. An extension where additional route and weather data are incorporated was developed in the deep-learning algorithm of [17]. However, the deep-learning module produces a deterministic forecast and neglects uncertainty of future driving speed. A stochastic velocity forecast based on explicitly modelled traffic phase information and modern navigation system data does not yet exist in literature, excluding the publication comprising preliminary results of this dissertation [111].

The VP algorithm proposed in this work uses feature-based forecasting. The features available can be accessed via the TRDB and consist of static route data from the geometric map and dynamic traffic data from connected probes. The used features are listed in Table 5.1.

Table 5.1: Features used for the regression algorithm provided by TRDB.

feature [unit]	details
$\hat{\phi}$ [-]	traffic phase
$\bar{u}$ [ms^{-1}]	ideal mean traffic speed
v_{lim} [ms^{-1}]	legal speed limit
κ [m^{-1}]	road curvature
Λ [-]	street class
$\bar{\alpha}$ [%]	mean road gradient angle

Static features v_{lim}, κ, Λ and $\bar{\alpha}$ are constant over time and depend only on the location. The only feature among these which is not self-explanatory is the street class Λ. It represents a categorization of the road segment according to the maximum capacity and speed. High volume and high speed roads (highways) are assigned class variable 1 and lower volume and/or lower speed roads are assigned higher classes up to 5 according to the TRDB definition [48]. The dynamic features $\hat{\phi}$ and $\bar{u}$ are not constant and depend on both the location and time. $\bar{u}$ is the ideal traffic speed explained in Section 5.1.1. Therefore, the forecast can be made for an ideal traffic prediction input, i.e. as if the dynamic traffic situation on the route is known beforehand. The traffic phase is not available from the TRDB today. However, given the probe data available in the back-end, this information can be extracted, for example based on the fundamental diagram using traffic speed, flow and density relationships [68, 136]. Hence, if this feature proves to help prediction accuracy, it could be transferred to the vehicle's control box to be used in the regression algorithm. A second possibility would be to equip connected vehicles with the traffic phase classification (TPC) algorithm developed in Section 4.2, thus enriching probe information on the TRDB with $\hat{\phi}$. In this work, the identified traffic phase on the measurement drive is used as predictive information, representing the ideal traffic phase forecast $\tilde{\phi}$. This is a coherent approach with the use of ideal traffic speed presented in Section 5.1.1.

5.1.3 Multi-Linear Regression

Apart from the features presented in the previous section, the regression algorithm itself must be selected. A simple model among candidate algorithms is the linear regression. It assumes a linear relationship between features and output variable $\bar{v}$. Using this linear model for multi-dimensional inputs requires multi-linear regression (MLR) which is the straightforward extension of the scalar model to a vector model [123, 3]. The connection, or *pattern*, describing the dependency of $\bar{v}$ on the features is expressed with the hypothesis vector $\mathbf{h}$. After the pattern has been identified using training data, the forecast can be made with help of of the predicted features for the selected route, which have been introduced in Section 5.1.2. Finding the hypothesis vector gives the following minimization problem:

$$\min_{\mathbf{h}} \sum_{k=1}^{n} \Theta\left(\bar{v}_k - \mathbf{h} \cdot \left[\bar{u}_k \; v_{\mathrm{lim},k} \; \phi_k \; \kappa_k \; \Lambda_k \; \bar{\alpha}_k\right]^{\top}\right) . \tag{5.1}$$

Obtaining optimal $\mathbf{h}$, which results in the smallest overall deviation from the measured speed $\bar{v}$ on all segments n of the training data, is dependent on the loss function Θ, a mathematical problem known from the parameter estimation (PE) in Section 4.1.1. The standard least-squares (LS) loss function would have the weight function $\Theta = (\cdot)^2$. However, LS is not robust, therefore in the MLR implementation for velocity prediction (VP) in this work, the Huber loss function is used for Θ to limit the outlier weights [52]. In addition, the observations are linearly weighted according to segment length l_k, hence longer segments are more important for the regression than shorter segments. To generate a forecast, hypothesis $\mathbf{h}$ from the training data and features from the future trip are required. Thereby the deterministic forecast $\tilde{v}^{\mathrm{d}}$ is the result of the direct computation $\mathbf{h}\,[\cdot]^{\top}$ following (5.1). For the stochastic forecast, the amount of uncertainty in the training data is included in the prediction, which is a common approach in forecasting applications [53]. Thereby, the relative error of the residuals is used to calculate the uncertainty. The predictive distribution is assumed to be Gaussian. This assumption is made to limit overfitting to the exact in-sample error distribution.

An additional challenge in forecasting velocity with MLR solely based on the input features is the lost time-dependency of the average velocity profile. At some points, that could lead to the prediction of implausible acceleration events. This could happen for example, if an on-ramp to a highway is part of the route, where traffic flows freely. The segment directly after the ramp will be predicted to have high velocity whereas the segment before has low speed because it is not on the highway yet. However, the ego-vehicle does not have infinite acceleration potential and the required acceleration to follow the predicted profile might be too high. Therefore, the resulting velocity prediction from the MLR is post-processed to remove such implausible acceleration events. For each driver, the acceleration behaviour is extracted from the training drive to calculate the personalized maximum and minimum acceleration. Subsequently, for each segment k of the predicted profile, the required acceleration a_{req} to reach $\tilde{v}$ on the next segment $k+1$ is calculated following (5.2). If a_{req} exceeds the personal maximum of the driver, the velocity prediction $\tilde{v}_{k+1}$ is corrected to be reachable within the acceleration limits. Exceeding personal minimum is treated equivalently.

$$a_{\text{req}} = \frac{v_{k+1}^2 - v_k^2}{2l_k} \tag{5.2}$$

To show an example for the VP using MLR, the ICEV test data set is used. As introduced in Section 3.4, the dataset contains two trips for each driver (A and B). Hence, one trip can be used for training the model and the trained model can subsequently be used to predict the return trip. The measurement data of the return trip can be used to evaluate the performance of the forecast. The VP algorithm is based on two model types, the regression model and the traffic phase classification (TPC) model, both must be defined. Selecting MLR for regression and the naive Bayesian classifier (NBC) from Section 4.2.2 for estimating ϕ results in the forecast signified with $\text{VP}_k\left(\tilde{v}_{mlr}, \hat{\phi}_{nbc}\right)$. The subscript k indicates that the forecast predicts a value for each segment, thus generating a profile instead of one aggregated quantity. In accordance with the used nomenclature in this work, a stochastic forecast generated by this model is denoted $\text{VP}_k\left(\tilde{v}^{\text{s}}_{mlr}, \hat{\phi}_{nbc}\right)$ and a deterministic forecast $\text{VP}_k\left(\tilde{v}^{\text{d}}_{mlr}, \hat{\phi}_{nbc}\right)$. Figure 5.6 shows an exemplary prediction of $\text{VP}_k\left(\tilde{v}_{mlr}, \phi_{nbc}\right)$. Thereby, the same driver and trip as Figure 5.5 is used, hence the profile for $\bar{v}$ is equal. The 90 % PI of the stochastic forecast $\tilde{v}^{\text{s}}$ is shown as well as the deterministic forecast $\tilde{v}^{\text{d}}$. In most cases, $\bar{v}$ is within the uncertainty band of the stochastic forecast. From the diagram, it seems the forecast does not systematically under-estimate future velocity as in the traffic speed prediction from Section 5.1.1. That means the tendency of the driver to choose higher speed than surrounding traffic was learned during the training trip. The magnitude of uncertainty in the forecast is higher compared to $\tilde{u}^{\text{s}}$. However, from this graph alone, it is not possible to verify how good the forecast is, or to compare different models with each other. Therefore, the reader is referred to the full analysis of the predictive performance given in Chapter 6.

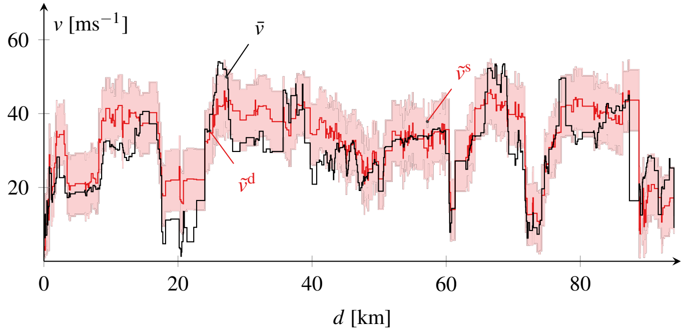
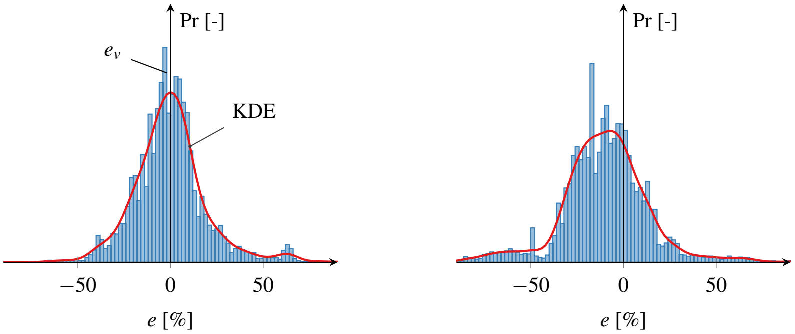


Figure 5.6: Velocity Prediction example $\mathrm{VP}_k\left(\tilde{v}_{mlr}, \hat{\phi}_{nbc}\right)$ using MLR as regression algorithm and NBC for estimating traffic phase on every segment k. The stochastic forecast is denoted $\tilde{v}^{\mathrm{s}}$ and the deterministic forecast $\tilde{v}^{\mathrm{d}}$. Thereby, $\tilde{v}^{\mathrm{s}}$ is shown as the 90 % PI with the shaded areas. The measured speed $\bar{v}$ is shown for comparison.

Figure 5.7 shows the in-sample and out-of-sample error distributions of the exemplary forecast $\mathrm{VP}_k\left(v_{mlr}, \phi_{nbc}\right)$ with the relative error

$$e_v = \left(\bar{v} - \tilde{v}^{\mathrm{d}}\right) \cdot \left(\tilde{v}^{\mathrm{d}}\right)^{-1} . \tag{5.3}$$

Although from the shown example alone, a general conclusion cannot be drawn, it indicates that the uncertainty is approximately uni-modal. The linear model is able to distinguish the traffic phases because of the feature input and therefore bi-modality previously encountered in the error distribution of the traffic speed prediction is not visible (see Section 5.1.1). When comparing in-sample and out-of-sample distribution, an ideal model would yield the same characteristics. In a forecasting application, this is hardly realistic, especially if the forecast is made far into the future and human behaviour is involved. Therefore the distributions should only be similar from a high level perspective to imply good generalization potential of the model. In this example, it appears the $\mathrm{VP}_k\left(\tilde{v}_{mlr}, \hat{\phi}_{nbc}\right)$ model is suitable for predicting vehicle speed. In addition, a significant amount of uncertainty can be observed which shows a deterministic forecast would not be sufficient in this case. However, how good the forecasts are and whether the defined assumptions hold can only be verified using more test drive data and a meaningful performance index, which is analysed in Chapter 6.



a) in-sample error distribution.

b) out-of-sample error distribution.

Figure 5.7: Example for the in-sample error distribution of speed prediction using $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{d}}_{mlr},\, \hat{\phi}_{nbc})$ for the above presented example. The kernel density estimate (KDE) with a Gaussian kernel is shown to visualize the characteristics of the distribution.

5.1.4 Support Vector Regression

In the previous section, a linear model was used for velocity prediction (VP). Since driving behaviour comes down to human behaviour, the system could also be non-linear which could not be represented well enough by the multilinear regression (MLR). Therefore, a more complex model should be implemented for comparison. Among candidate algorithms for pattern analysis, the support vector machine (SVM) has been proven to be a suitable algorithm for travel time prediction by [140], which is a very similar application. In a traffic speed forecast project [130], the performance of an SVM was compared to a neural network (NN), which is one of the most common machine learning algorithms nowadays. It was shown that SVM produced more accurate results, especially when dealing with limited training data quality and quantity. Based on these assessments, where the features were limited to probe data, SVM was chosen as candidate machine learning algorithm for the velocity prediction (VP) algorithm implemented in this work. The SVM was originally created to solve non-linear classification problems by V. Vapnik [132, 131] and uses the concept of separating hyperplanes as illustrated in Figure 5.8. There, it can be seen that the optimal hyperplane to separate (classifies) red and blue points maximizes the margin, i.e. maximizes the distance to the nearest points. With mathematical optimization algorithms, the hyperplane can be found for an arbitrary set of points to be classified [131]. However, in this example it can be seen that the methodology works only for linear problems.

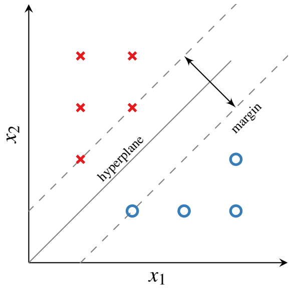


Figure 5.8: Illustration of the separation mechanism for SVMs. The optimal hyperplane to separate blue and red points has the maximum margin following [131].

To be suitable for non-linear problems, the SVM uses the *kernel transformation* to transform a problem which has no linear solution in the normal input space into a higher dimensional space where a linear solution exists. An illustration of this methodology can be seen in Figure 5.9. There, classification of the red and blue points in the left diagram is not possible with a linear hyperplane, as it is an ellipsoid. Using a quadratic transformation function

$$\zeta(x_1,x_2) = [x_1^1, \sqrt{2}x_1x_2, x_2^2] \tag{5.4}$$

the points can be transferred to a higher dimensional feature space ζ, where the points are linearly separable as indicated with the dotted rectangle. The transformation into the feature space can be described with a Kernel function $K(x_1,x_2)$, in the above case it would be a quadratic Kernel model. Kernel functions are designed in a specific way, such that they enable a computationally efficient transformation, as shown in [115]. Indeed, it can be mathematically shown that using the right Kernels, the calculation of the separating hyperplane for non-linear problems is still of linear computational complexity although a transformation into a higher-dimensional feature space is inherent. To comprehend this mathematical trick, also called the *kernel trick*, the reader is referred to the text books about SVM, e.g. [112, 115] because computational efficiency is not within the focus of this work.

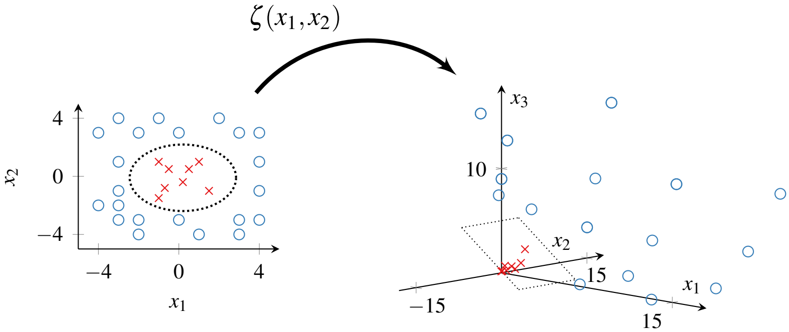


Figure 5.9: Exemplary transformation into a higher dimensional feature space. The left diagram shows the points to be separated in 2D and the right diagram illustrates the transformed points in the three dimensional feature space. Thus, the linearly inseparable points in 2D become linearly separable in 3D.

SVM can also be used for regression with a slightly changed mathematical methodology. Instead of maximizing the margin between separable data points, in regression, the optimal hyperplane minimizes the distance to the data points. As introduced in Section 4.1 and Section 5.1.3, a loss function is required to find this hyperplane, in this case Vapnik's ε-insensitive loss function [131]. Thereby, a margin ε is selected within an error is not relevant and points outside the margin are weighted linearly, as illustrated in Figure 5.10. Hence, as in the classification case, the distance of the data points to the margin is the value to be optimized. In the work of [118], it was shown that the ε-insensitive loss function leads to the equivalent optimization function as in the classification case. Therefore, the kernel transformation can be used to solve non-linear regression problems. In consequence, SVM based systems can solve non-linear dependencies which MLR cannot. This is an advantage, as it cannot be assumed that driving behaviour follows linear equations. In this work, the SVM implementation of [118] was used with a polynomial kernel with the same features as the MLR, therefore the reader is referred to the literature for more details about the algorithm.

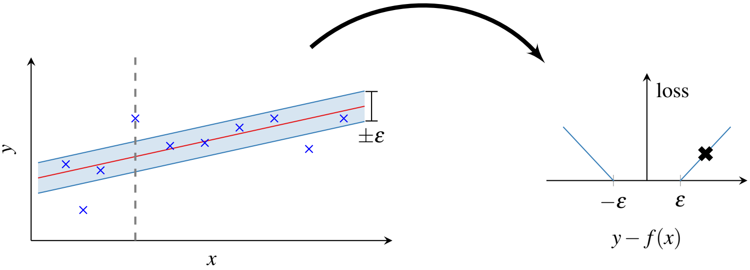


Figure 5.10: Illustration of the SVM method for regression using the ε-insensitive loss function following [112]. On the left, the data points with the regression line and the ε-insensitive band is visible. On the right, the loss for the selected point is calculated.

Equivalently to the MLR model presented in the previous section, the SVM model cannot capture the time dependency as it is a feature-based forecast.

Therefore, the prediction must be post-processed to ensure a plausible acceleration behaviour. For this, the methodology of MLR is used, see Section 5.1.3 for details. In addition, the linear weighting on the segment length is included in the ε-insensitive loss function, as in the MLR. Standard SVM implementations generate deterministic forecasts, but for a stochastic forecast, the uncertainty of the model must be included. Following [53], the use of the in-sample error distribution as out-of-sample uncertainty is often a viable and straightforward alternative in forecasting applications. This method is applied in the linear case and therefore also for the SVM forecast. Figure 5.11 shows the prediction of the SVM model for the same training and test data as in the examples of Section 5.1.1 and Section 5.1.3. It uses the traffic phase estimates from the naive Bayesian classifier (NBC) and is consequently denoted $\mathrm{VP}_k\left(\tilde{v}_{svm},\, \hat{\phi}_{nbc}\right)$. It can be seen that the SVM forecast obtains plausible results for the presented test drive and the measurement $\bar{v}$ is mostly within the 90 % PI. Even the peak at around 30 km is within the uncertainty distribution, which is an evidence of the SVM to learn driving behaviour from the training set. Interestingly, the low speed phase at around 20 km is under-estimated whereas it was over-estimates in the MLR forecast. However, for a reliable quantitative evaluation of all approaches, the reader is referred to Chapter 6.

Figure 5.12 shows the in-sample and out-of-sample error distributions of the model $\mathrm{VP}_k\left(\tilde{v}^{\mathrm{d}}_{svm},\, \hat{\phi}_{nbc}\right)$ with the relative error calculated according to (5.3). Hereby, similar characteristics to the MLR error distributions can be detected. An ideal forecast would have the same in-sample and out-of-sample error distribution. In this application this is unrealistic because of too many influence factors, one of them being human behaviour. The KDE reveals a uni-modal in-sample error, therefore the bi-modality due to different traffic phases is not visible. From this observation, it is expected that forecast algorithms including an estimate for traffic phase generate better results than algorithms neglecting it. The out-of-sample error shows slight bi-modality but the modes are very close to each other. In addition, a significant amount

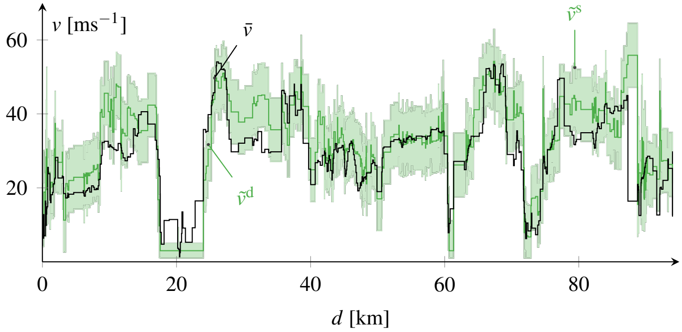
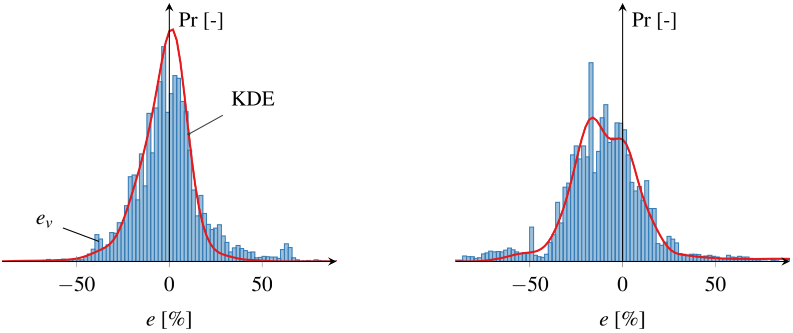


Figure 5.11: Velocity Prediction example $\mathrm{VP}_k\,(\tilde{v}_{svm},\,\hat{\phi}_{nbc})$ using SVM as regression algorithm and NBC for estimating traffic phase on every segment k. The stochastic forecast is denoted $\tilde{v}^{\mathrm{s}}$ and the deterministic forecast $\tilde{v}^{\mathrm{d}}$. Thereby, $\tilde{v}^{\mathrm{s}}$ is shown as the 90 % PI with the shaded areas. The measured speed $\bar{v}$ is shown for comparison.

of uncertainty can be observed, an indication that deterministic forecasting is not suitable in this application. On a high level, it appears the SVM model is adequate for predicting vehicle speed. This hypothesis can only be verified using more test drive data, which is analysed in Chapter 6.

5.2 Energy Consumption Forecast

After obtaining the velocity prediction (VP) presented in Section 5.1, the future energy consumption $\tilde{E}_{\mathrm{c}}$ can be calculated following the overall methodology shown in Figure 5.1. In the deterministic case, this could be achieved by directly using the energy consumption model of the vehicle from Section 4.1.2 and the future velocity profile $\tilde{v}^{\mathrm{d}}$, resulting in the prediction $\tilde{E}_{\mathrm{c}}^{\mathrm{d}}$. However, in the stochastic case the uncertainty of $\tilde{v}^{\mathrm{s}}$ makes a direct computation more complex because the effect of the uncertainty distribution on the energy consumption must be assessed. This problem is called *uncertainty propagation* and different common approaches are shown in Section 5.2.1. In this work, the sequential Monte-Carlo simulation (MCS) is chosen among the candidates and is introduced in Section 5.2.2. After thus obtaining the



a) in-sample error distribution.

b) out-of-sample error distribution.

Figure 5.12: Example for the in-sample and out-of-sample error distributions of speed prediction using $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{d}}_{svm}, \hat{\phi}_{nbc})$ for the above presented example. The kernel density estimate (KDE) with a Gaussian kernel is shown to visualize the characteristics of the distribution.

stochastic energy consumption forecast $\tilde{E}^{\mathrm{s}}_{\mathrm{c}}$, the overall result must be calculated by convolving all the distributions for the predicted segments, which is introduced in Section 5.2.3.

5.2.1 Approaches for Uncertainty Propagation

Uncertainty propagation methods can assess the uncertainty of future energy consumption $\tilde{E}_{\mathrm{c}}$ resulting from uncertain $\tilde{v}$, which is required for calculating attainability based on stochastic forecasts. In a general description, uncertainty propagation covers the effect of an uncertain input variable x on a system and thus on the output variable z, as shown in Figure 5.13. Therein, the system is described with the *transfer function* $f(x)$. In the ECF context, x would be the $\tilde{v}^{\mathrm{s}}$ distribution, $f(x)$ the energy consumption model shown in Section 4.1 and z the $\tilde{E}^{\mathrm{s}}_{\mathrm{c}}$ distribution. For systems where the distributions and the transfer function can be described analytically, the uncertainty propagation is directly computable. However, real systems are often analytically intractable. This is also the case in the presented application since the transfer function is non-linear. To solve this problem, multiple methodologies were developed by researchers to approximate the resulting distribution z or the transfer function $f(x)$.

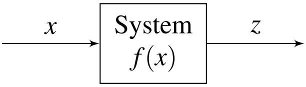


Figure 5.13: Block diagram for one dimensional uncertainty propagation. The uncertainty of an input variable x effects the system and consequently the uncertainty of an output variable z. Thereby, the system is described with the transfer function $f(x)$.

A commonly known form of uncertainty propagation is the problem of *error propagation* [79]. Sometimes, it is also called *method of moments*, as it estimates the effects of an input error in terms of its first and second statistical moments, which is mean and variance. It is based on an approximation of the transfer function and can be calculated using the following formula, extended to the two-dimensional case:

$$s_z \approx \sqrt{\left(\frac{\partial f}{\partial x} s_x\right)^2 + \left(\frac{\partial f}{\partial y} s_y\right)^2 + 2r_{xy}\left(\frac{\partial f}{\partial x} s_x\right)\left(\frac{\partial f}{\partial y} s_y\right)} \,. \tag{5.5}$$

There, r_{xy} is the correlation between the input variables. Using this formula for ECF with the only uncertain input being the velocity profile gives

$$s_E \approx \sqrt{\left(\frac{\partial E_\mathrm{c}}{\partial v} s_v\right)^2} \,, \tag{5.6}$$

where s_E is the uncertainty in energy consumption and s_v the uncertainty in the velocity. The two quantities are analytically connected via the energy consumption model, here denoted with E. Constructing the local derivative of $E(v)$ in the point of interest, the linearisation of the transfer function allows direct computation of s_E as shown in Figure 5.14.

Usually, measurement errors are assumed to have Gaussian uncertainties, hence in some sources the method is also called *Gaussian error propaga-*

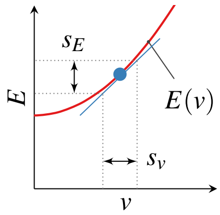


Figure 5.14: Illustration of the Gaussian error propagation method. The error s_E is obtained using the local derivative of the function $E(v)$ at the point of interest and the original error s_v.

tion. It is widely used but in addition to assuming Gaussian uncertainties, it also requires negligible linearisation errors. Oftenly, real systems do not fulfill these preconditions which could lead to biased results as shown by [4]. In this work, an additional challenge lies in the transfer function itself since the energy consumption model is based on characteristic maps, hence a function approximation of the model would be required which could induce additional bias.

Another approach for uncertainty propagation where the transfer function is not approximated or linearised is the *unscented transformation* originally proposed by [60]. It was created for signal processing problems of non-linear systems. Such systems cannot be solved with the linear Kalman filter, therefore the extended Kalman filter (EKF) was created which linearises the transfer function, similar to the above mentioned error propagation. [60] proposed an alternative where instead of approximating the transfer function in the EKF, the uncertainty distributions are approximated. Thereby, a set of sample points is chosen and propagated through the system using the exact transfer function. In opposition to Monte-Carlo methods, these sample points are not selected randomly but follow a deterministic algorithm. With this method, the first and second moment of the output can be calculated based on a small number of points. In their work [62, 61, 59], the authors proved that unscented transformation is superior to the linearisation approach in many real-world applications. Still, the in- and output uncertainties are assumed to be near-Gaussian.

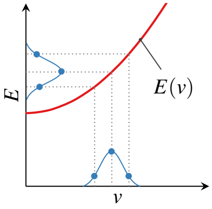


Figure 5.15: Illustration of the unscented transformation. Instead of approximating the transfer function, the distributions are approximated using a set of sample points.

However, for an application in this work, the kinetic energy change between the segments must be considered for calculating energy consumption following the model presented in Section 4.1. Therein lies an additional challenge for the uncertainty propagation method: The kinetic energy depends on the velocity profile which is subject to uncertainty, as illustrated in Figure 5.16. There, it is shown that the total energy consumption depends on the sequence of the predicted distribution over the segments k. Theoretically, the kinetic energy change necessary for all points on k and $k+1$ within the predicted distribution could be calculated. However, not all the states are plausible or physically possible and the computation time would be very high. Therefore, not only a method which is able to propagate the uncertainty through the non-linear system but also a model for the *sequentiality* of the future profile is required. Since the introduced methods have limited capability to solve this problem, a popular approach is the sequential Monte-Carlo simulation (MCS) explained in the next section.

5.2.2 Sequential Monte-Carlo Simulation

A common method for uncertainty propagation used in research is the Monte-Carlo simulation [88]. It approximates the input distribution with

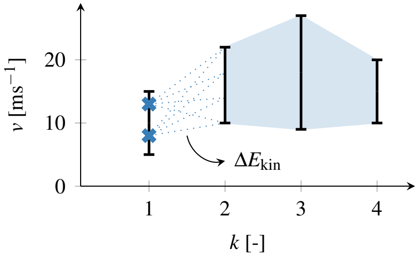


Figure 5.16: Illustration of the sequentiality in uncertainty propagation for energy consumption. The kinetic energy change ΔE_{kin} is dependent on the uncertain velocity profile.

a set of arbitrary points, which are propagated through the transfer function to form the output distribution, as illustrated in Figure 5.17. The main difference to the unscented transformation shown in the previous section is the method of choosing the input points, since it does not follow a deterministic algorithm. A Monte-Carlo simulation does not impose assumptions over the input- and output distributions nor does it approximate the transfer function.

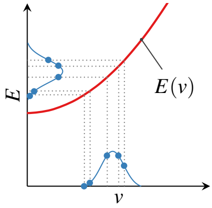


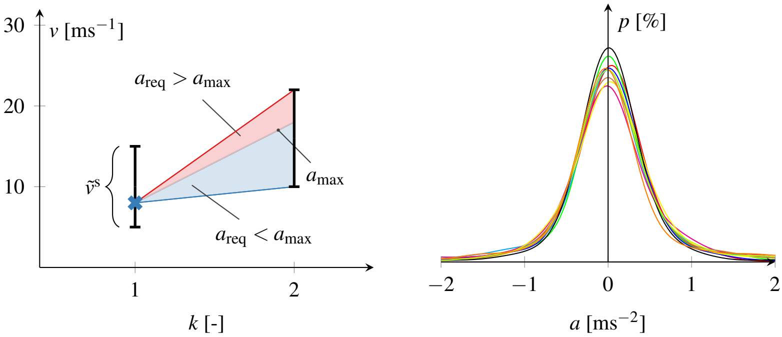
Figure 5.17: Illustration of the Monte-Carlo Simulation Method. A set of arbitrary points is selected to approximate the input distribution. Each point is propagated through the transfer function to form the output distribution.

At the start of a Monte-Carlo simulation, the points to represent the input distribution must be chosen. These points are called *samples*, consequently the procedure of selecting them is the *sampling method*. There are various sampling methods, see [58] for an overview. In this work, two of them are presented: uniform sampling and importance sampling. In uniform sampling, the n random samples $\{\xi^{(i)}\}_{i=1}^{n}$ are drawn from a uniform distribution. The number of samples n is chosen based on the trade-off between computation time and accuracy. It can be imagined that an extremely high number of samples approximates the input distribution very accurately but also increases calculation time significantly since each sample must be evaluated. If the input distribution and the sampling distribution are very different, more samples are required for adequate accuracy than if the distributions are similar.

Hence, uniform sampling might not lead to acceptable results because the input distribution in real-world applications rarely follows a uniform distribution. Thus, a generalization of uniform sampling, called *importance sampling* can be used. In importance sampling, the user specifies a distribution Q from which the samples are drawn, e.g. a Gaussian. In consequence, the samples $\{\xi^{(i)}\}_{i=1}^{n}$ are more likely to be chosen from an area with high probability in Q which leads to a better approximation with less samples if the input distribution and Q are similar.

The standard Monte-Carlo simulation can be extended to cover dependencies between the states in the so called sequential Monte-Carlo simulation (MCS) [15]. In MCS, a number of particles τ is specified and each particle is sequentially propagated through the transfer function. Thereby, in each sequence, a random sample is chosen following the specified sampling method. This enables MCS to solve the uncertainty propagation problem posed by the necessity to forecast energy consumption based on an uncertain velocity profile.

As shown in Section 5.2.1, the individual velocity samples cannot be chosen independently of each other. While in Section 5.1.3 and Section 5.1.4, the deterministic prediction was post-processed to be realistic, the stochastic forecast was not. However, it can be imagined that, if a velocity sample close to the lower boundary of the PI is chosen on segment k, not all velocity values within the PI of segment $k+1$ are accessible with reasonable acceleration. This is illustrated in Figure 5.18a, where the blue cross signifies the chosen velocity sample on k within the stochastic velocity prediction $\tilde{v}^{\mathrm{s}}$. The red region illustrates the implausible velocity values which exceed the physical limit of the test vehicle, as the powertrain cannot provide enough torque if the required acceleration a_{req} is above the maximum acceleration $a_{\max}$. Furthermore, among the physically plausible accelerations, not all are equally likely. Driver's acceleration behaviour usually follows a Gaussian distribution, as shown by [108]. Drivers on long-distance trips especially will tend to a calm driving style with more low than high accelerations. To evaluate this, the acceleration behaviour of all 10 test drivers in the ICEV data set was analysed and the KDEs are shown in Figure 5.18b. It can be seen that the distributions do not differ significantly between the drivers and all of them rarely accelerate or decelerate with more than $1\,\mathrm{m\,s^{-2}}$. These distributions can be used for the sampling method in order to achieve a realistic velocity profile for each particle i.



a) Dependency of $\tilde{v}_{k+1}$ on $\tilde{v}_k$.

b) Acceleration behaviour of the test drivers.

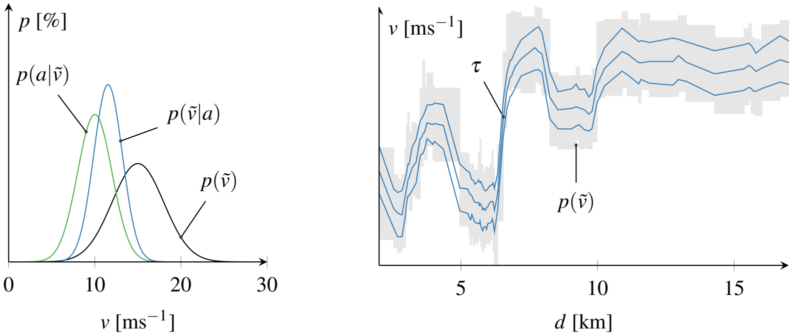
Figure 5.18: Illustration of the inter-dependency of the velocity samples for MCS. Figure 5.18a shows the dependency of the velocity uncertainty on a segment $k+1$ on the prior segment k via the physical acceleration limit $a_{\max}$. Figure 5.18b shows the acceleration kernel density estimation (KDE) of the 10 test drivers.

The sampling distribution Q can be modelled with a Bayesian approach where the distribution given by the VP serves as prior distribution $p(\tilde{v}) = \tilde{v}^{\mathrm{s}}$. The likelihood distribution represents probable individual acceleration behaviour of the driver $p(a|\tilde{v})$ given the VP, following Figure 5.18b. The posterior distribution $p(\tilde{v}|a)$, which reflects both the velocity profile and the dependency of each sample on the previous one, can subsequently be calculated with Bayes' rule:

$$p(\tilde{v}|a) = \frac{p(a|\tilde{v})p(\tilde{v})}{p(a)} \; . \tag{5.7}$$

An exemplary illustration is shown in Figure 5.19a, where the prior $p(\tilde{v})$, likelihood $p(a|\tilde{v})$ and the resulting posterior $p(\tilde{v}|a)$ are shown. That posterior distribution serves as sampling distribution Q for the importance sampling method presented above. The individual velocity samples $\{v^{(k,i)}\}_{k,i=1}^{n,\tau}$ are subsequently drawn for each particle i and on each segment k from the posterior distribution. An exemplary trajectory of three particles can be seen in Figure 5.19. There, no unrealistically high accelerations are visible and the trajectories have a similar qualitative course but differ due to their stochastic nature.

The energy consumption of each particle on each segment can be calculated using the model presented in Section 4.1. The stochastic forecast for $\tilde{E}^{\mathrm{s}}_{\mathrm{c},k}$ is afterwards obtained by collecting the energy and probability of each sample on the specific segment. Thus, the desired result of the uncertainty propagation problem is achieved. An example is shown in Figure 5.20. There, $\tau = 10$ particles were used which means a total number of 4320 velocity samples $\{v^{(k,i)}\}_{k,i=1}^{n,\tau}$ on $n = 432$ segments must be evaluated. The chosen VP algorithm used a naive Bayesian classifier (NBC) for identifying the traffic phase and multi-linear regression (MLR) for learning the driver behaviour $\mathrm{VP}_k\,(\tilde{v}_{mlr}, \hat{\phi}_{nbc})$. The 90 % PI of the stochastic forecast $\tilde{E}^{\mathrm{s}}_{\mathrm{c}}$ is illustrated with the shaded area and the observation from the measurement on each segment is denoted $\bar{E}_{\mathrm{c}}$. From the graph, it seems the observation is mostly within



a) Probability distributions used for the sampling method

b) Exemplary velocity profile of three particles τ

Figure 5.19: Exemplary illustration of the sequential Monte-Carlo simulation (MCS). Figure 5.19a shows how the sampling distribution is computed using Bayes' rule. From the resulting posterior, the particles τ select velocity samples on each segment. The resulting profile of three particles is shown in Figure 5.19b.

the prediction interval. However the overall performance of the forecast is evaluated in Chapter 6.

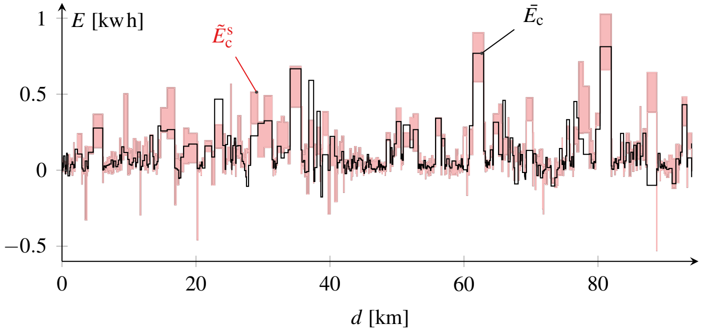


Figure 5.20: Energy consumption forecast resulting from the sequential Monte-Carlo simulation (MCS) with 10 particles and importance sampling based on the velocity prediction model $\mathrm{VP}_k\,(\tilde{v}_{mlr}, \hat{\phi}_{nbc})$.

5.2.3 Convolution

The output of the sequential Monte-Carlo simulation (MCS) is the the individual energy consumption distribution on every segment $\tilde{E}^{\mathrm{s}}_{\mathrm{c},k}$. To achieve the total energy consumption forecast $\tilde{E}^{\mathrm{s}}_{\mathrm{c}}$, the sum of these individual distributions must be obtained. For deterministic predictions, this would be the sum of the predicted energy on all segments k. For stochastic forecasts, the predicted energy is a random variable on each k. The sum of random variables can be obtained using *convolution* [49]. Normally, convolution requires linear independence. While the predictions on the individual segments are independent on each other as output of the VP, the result is post-processed to exclude unrealistic accelerations. This post-processing is regarded as weak dependence. In addition, the segments are only pairwise dependent which is another definition for weak dependence. In [110], it was shown that convolution can be used for weak dependent systems.

The analytical formulation of convolution is given in the following. Let the resulting distribution for two segment energies $\tilde{E}^{\mathrm{s}}_{\mathrm{c},1}$ and $\tilde{E}^{\mathrm{s}}_{\mathrm{c},2}$ be given by $\tilde{E}^{\mathrm{s}}_{\mathrm{res},2}$. Then, for an arbitrary point y, the resulting distribution can be calculated:

$$\tilde{E}^{\mathrm{s}}_{\mathrm{res},1} = \tilde{E}^{\mathrm{s}}_{\mathrm{c},1} \tag{5.8}$$

$$\begin{aligned} \tilde{E}^{\mathrm{s}}_{\mathrm{res},2}(y) &= \tilde{E}^{\mathrm{s}}_{\mathrm{res},1} * \tilde{E}^{\mathrm{s}}_{\mathrm{c},2} \\ &= \int \tilde{E}^{\mathrm{s}}_{\mathrm{res},1}(z_1) \cdot \tilde{E}^{\mathrm{s}}_{\mathrm{c},2}(y - z_1) dz_1 \end{aligned} \tag{5.9}$$

Extending (5.9) to the general case finally gives

$$\tilde{E}^{\mathrm{s}}_{\mathrm{res},k+1} = \tilde{E}^{\mathrm{s}}_{\mathrm{res},k} * \tilde{E}^{\mathrm{s}}_{\mathrm{c},k+1} \; \forall k \geq 1 \,, \tag{5.10}$$

$$\tilde{E}^{\mathrm{s}}_{\mathrm{c}} = \tilde{E}^{\mathrm{s}}_{\mathrm{c},1} * \ldots * \tilde{E}^{\mathrm{s}}_{\mathrm{c},n} = \tilde{E}^{\mathrm{s}}_{\mathrm{res},n} \,. \tag{5.11}$$

With (5.11), the total energy consumption for all n segments can be calculated, which is the goal of the energy consumption forecast (ECF). Due to the central limit theorem (CLT), the resulting distribution for $\tilde{E}^{\mathrm{s}}_{\mathrm{res},n} = \tilde{E}_{\mathrm{c}}$ is normally distributed. The CLT states that the sum of random variables which are not Gaussian tends to a normal distribution and is based on the *law of large numbers* [3, 22]. In [108], it was shown that even on a small number of segments and a bi-modal distribution, the resulting energy consumption from the convolution is normally distributed. The same can be shown using an exemplary VP with SVM neglecting TPC, denoted $\mathrm{VP}_k\left(\tilde{v}^{\mathrm{s}}_{svm}\right)$. Figure 5.21 illustrates different steps in the convolution of the MCS results. The convolution of the first two segments $\tilde{E}^{\mathrm{s}}_{\mathrm{res},2}$ is strongly non-normal but with the convolution of more segments, the resulting uncertainty distribution becomes Gaussian after 25 segments due to the central limit theorem (CLT). In this example, the 25 segments represent a prediction of 2.5 km into the future. From this, it can be concluded that in a range estimation context, the amount of predicted segments is always enough for the resulting distribution to be normal.

Convolving all predictions $\tilde{E}^{\mathrm{s}}_{\mathrm{c},k}$ results in the total energy consumption forecast $\tilde{E}^{\mathrm{s}}_{\mathrm{c}}$, which can subsequently be used to calculate attainability. Figure 5.22 shows an example of such a forecast. There, the stochastic forecast based on the probe distribution for velocity prediction from Section 5.1.1 is denoted $\tilde{E}^{\mathrm{s}}_{\mathrm{c}}(\tilde{u})$. The result obtained using the same regression model $\mathrm{VP}_k\left(\tilde{v}^{\mathrm{s}}_{mlr}, \hat{\phi}_{nbc}\right)$ for velocity prediction as in Figure 5.20 is denoted $\tilde{E}^{\mathrm{s}}_{\mathrm{c}}(\tilde{v})$. For the required MCS, 10 particles are used in both cases. As a comparison, also the deterministic alternatives based on the mean values from velocity prediction $\tilde{E}^{\mathrm{d}}_{\mathrm{c}}(\tilde{u})$ and $\tilde{E}^{\mathrm{d}}_{\mathrm{c}}(\tilde{v})$ are shown. Thus, they are not the maximum a posteriori (MAP) estimates of the respective stochastic forecasts. The observed energy consumption on this test drive is denoted E_{c}. In this example, it can be seen both deterministic predictions under-estimate energy consumption for the future trip, as E_{c} is 1.59 kWh (7.00 %) higher than $\tilde{E}^{\mathrm{d}}_{\mathrm{c}}(\tilde{u})$ and 0.47 kWh (1.97 %) higher than $\tilde{E}^{\mathrm{d}}_{\mathrm{c}}(\tilde{v})$. Looking at the stochastic forecasts, it can be seen that the realization based on the test-drive measurement E_{c} is well within the predicted uncertainty distributions. The expected

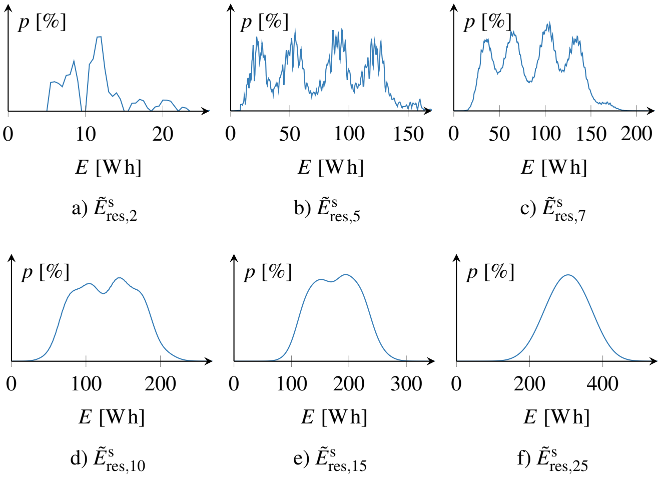


Figure 5.21: Illustration of different steps in the convolution for an exemplary energy consumption forecast. $\tilde{E}^{\mathrm{s}}_{\mathrm{res},2}$ shows the result of the first and second segment prediction added together, $\tilde{E}^{\mathrm{s}}_{\mathrm{res},5}$ the first five segments etc. After the 25th segment, the resulting uncertainty distribution is a Gaussian due to the central limit theorem (CLT).

values $\mathbb{E}(\tilde{E}^{\mathrm{s}}_{\mathrm{c}}(\tilde{u})) = 23.61\,\mathrm{kWh}$ and $\mathbb{E}(\tilde{E}^{\mathrm{s}}_{\mathrm{c}}(\tilde{v})) = 24.36\,\mathrm{kWh}$ of the stochastic forecasts are closer to $E_{\mathrm{c}} = 24.32\,\mathrm{kWh}$ than their respective deterministic counterparts. In addition, the predictions using the regression model are superior to the probe information predictions. The question whether this is true for all test drives will be answered in Chapter 6.

5.3 Attainability Calculation

After completing the energy consumption forecast (ECF) in the previous section, the attainability p_{a} can be calculated following the methodology presented in Figure 5.1. Other than the predicted energy consumption $\tilde{E}_{\mathrm{c}}$, the estimated battery energy content $\hat{E}_{\mathrm{b}}$ is required for the calculation. $\hat{E}_{\mathrm{b}}$ can be obtained using the battery model shown in Section 3.3 where en-

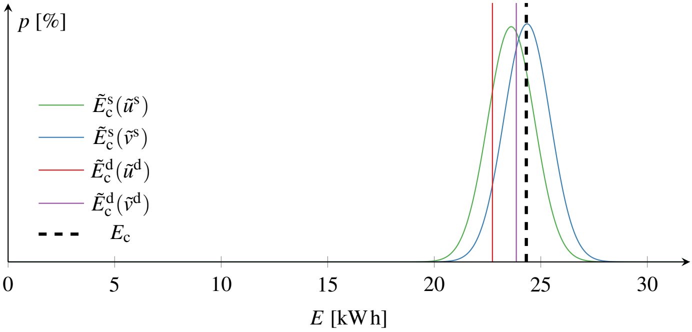


Figure 5.22: Total convoluted energy consumption forecast on an exemplary test drive. The stochastic forecasts are based on the probe distribution $\tilde{E}_c^s(\tilde{u}^s)$ and on the regression model $\mathrm{VP}_k\,(\tilde{v}^s_{mlr}, \hat{\phi}_{nbc})$, denoted $\tilde{E}_c^s(\tilde{v}^s)$. The deterministic forecasts $\tilde{E}_c^d(\tilde{u}^d)$ and $\tilde{E}_c^d(\tilde{v}^d)$ use mean values from the respective velocity prediction. E_c represents the real required energy from the measurement.

ergy content uncertainty originates from SOC and SOH measurement uncertainty. In addition, a mean degree of efficiency of $\eta = 98\,\%$ is assumed. For realistic battery parameters, the characteristics of the BEV test vehicle battery presented in Section 3.4 are applied in the model. This battery uses pouch cells ($s = 100, p = 4$), has a nominal voltage of 370 V and a peak power of 270 kW. An application and detailed analysis of the battery model and its parameters were already published in [108]. For the calculation of attainability in this work, the important output of the model is the distribution of $\hat{E}_b$ depending on SOC (measured in q) and SOH (measured in h), which is illustrated in Figure 5.23. There, a new battery $\hat{E}_{b,\text{new}}$ with $h = 1$ and an aged battery $\hat{E}_{b,\text{aged}}$ with $h = 0.8$ is shown. It can be seen that the battery energy content has significant uncertainty and neglecting it could lead to a biased range estimation. Usable energy decreases with battery age and uncertainty increases compared to a new battery. If a deterministic estimation of battery energy content is used in the following of this work, it represents the MAP of the uncertainty distributions, indicated with the solid lines in Figure 5.23.

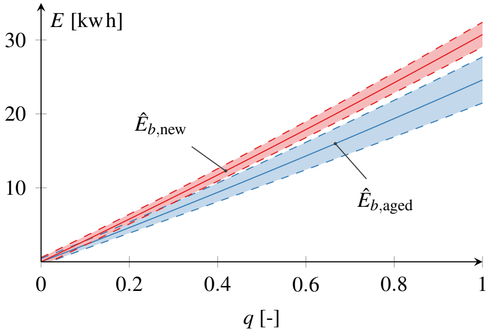


Figure 5.23: Estimate of the battery energy content with the battery model. The mean values and uncertainties for a new battery $\hat{E}_{b,\text{new}}$ and an aged battery $\hat{E}_{b,\text{aged}}$ are shown where the shaded area represents 2σ of the uncertainty distribution.

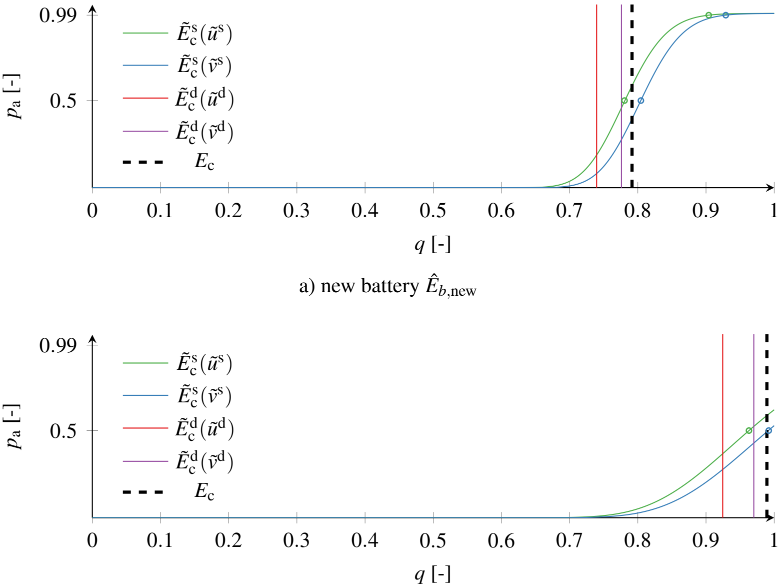
Taking the battery model and the exemplary ECF from Section 5.2.3, attainability p_a can be calculated over the whole SOC range following the method from Section 3.2. Thereby, the probability of $\hat{E}_\text{b}$ being greater or equal than $\tilde{E}_\text{c}$ can be evaluated using the approach of [86] with the mean values and standard deviations of the respective distributions as well as the cumulative density function of the standard normal distribution Φ:

$$p_\text{a} = 1 - \Phi\left(-\frac{\mu_{\hat{E}_\text{b}} - \mu_{\tilde{E}_\text{c}}}{\sqrt{\sigma_{\hat{E}_\text{b}} + \sigma_{\tilde{E}_\text{c}}}}\right) . \tag{5.12}$$

Calculating (5.12) for the same trip and prediction models as in Figure 5.22 gives the results shown in Figure 5.24. Thereby, $\hat{E}_\text{b}$ was obtained for all possible q and thus also p_a. In Figure 5.24a, a new battery scenario is used and in Figure 5.24b, an aged battery is used. The denotation and the underlying prediction models are equivalent to the results in Figure 5.22, e.g. the SOC required to supply the measured energy demand is denoted with E_c. The deterministic algorithms also use deterministic $\hat{E}_\text{b}$ estimation whereas the stochastic forecast includes the uncertainty of the battery energy content.

Looking at the new battery scenario, similar characteristics to the ECF results can be seen. The deterministic predictions under-estimate the necessary charge, which could lead to unpredicted stranding. Among the two, the forecast using the regression model $\tilde{E}_{\mathrm{c}}^{\mathrm{d}}(\tilde{v}^{\mathrm{d}})$ is closer to the measurement than the forecast using mean traffic speed $\tilde{E}_{\mathrm{c}}^{\mathrm{d}}(\tilde{u}^{\mathrm{d}})$. Especially the former forecast seems to be very accurate but since it is a deterministic forecast, knowledge about accuracy was not transported in the prediction. Analysing the stochastic forecasts with probe distribution $\tilde{E}_{\mathrm{c}}^{\mathrm{s}}(\tilde{u}^{\mathrm{s}})$ and the regression model $\tilde{E}_{\mathrm{c}}^{\mathrm{s}}(\tilde{v}^{\mathrm{s}})$, it can be seen that the realization E_{c} is within the predicted distribution. With the resulting curves of p_{a}, the user could now specify the probability of arrival. Selecting $p_{\mathrm{a}} = 0.5$ would mean that the forecast calculated a 50 % chance of arriving at the destination with the respective SOC. Figure 5.24b illustrates the forecast for the same trip but with an aged battery. The ECF results remain unchanged but the battery energy content $\hat{E}_{\mathrm{b}}$ is reduced and its uncertainty increased, as previously shown in Figure 5.23. It can be seen that overall higher required SOC values are predicted than in the new battery scenario, which is a logical consequence from having less energy in the battery. Also, the curves for stochastic attainability are shallower due to increased uncertainty. It stands out that stochastic attainability values do not exceed 50 % by much, which indicates a high stranding risk.

The quantitative results for the new battery scenario are illustrated in Table 5.2. Since the deterministic forecasts do not include a measure of uncertainty, they have only solutions for $p_{\mathrm{a}} = 1$. The stochastic forecasts both do not reach $p_{\mathrm{a}} = 1$ for a full battery, thus the possibility of higher energy demand than energy content cannot be discarded absolutely. However, accepting 1 % risk leads to quantitative solutions, e.g. $\tilde{E}_{\mathrm{c}}^{\mathrm{s}}(\tilde{u}^{\mathrm{s}})$ indicates $q = 0.904$ is enough charge for $p_{\mathrm{a}} = 0.99$. Furthermore, for $p_{\mathrm{a}} = 0.5$, a charge of $q = 0.781$ is predicted to be sufficient, which in this case would be below the real required value E_{c} and therefore the vehicle would have been stranded. Taking the forecast $\tilde{E}_{\mathrm{c}}^{\mathrm{s}}(\tilde{v}^{\mathrm{s}})$, even $p_{\mathrm{a}} = 0.5$ would have been enough energy to complete the trip. To ensure safely arriving at a destination, drivers re-



a) new battery $\hat{E}_{b,\text{new}}$

b) aged battery $\hat{E}_{b,\text{aged}}$

Figure 5.24: Attainability on an exemplary test drive using the battery model. The stochastic forecasts are based on the probe distribution $\tilde{E}_c^s(\tilde{u}^s)$ and on the regression model using NBC and MLR $\tilde{E}_c^s(\tilde{v}^s)$. The deterministic forecasts use only mean velocity prediction without uncertainty intervals. E_c represents the real required SOC from the measurement. Figure 5.24a shows attainability when using a new battery and Figure 5.24b illustrates the use of an aged battery. Positions for $p_a = 0.99$ and $p_a = 0.5$ are indicated by the small points on the curves.

serve *safety margins*, as shown by [33]. In typical range estimation applications, forecasts are executed deterministically and constant safety margins are added, e.g. $q = 0.2$ in [18]. In stochastic range prediction, it can be evaluated numerically.

The quantitative results for the aged battery scenario are illustrated in Table 5.3. It can be seen that for an aged battery, having enough energy in the battery to complete the trip E_c means $q = 0.989$ which is almost the full battery capacity. As in the new battery scenario, both deterministic fore-

Table 5.2: Quantitative results for attainability in the new battery scenario $\hat{E}_{b,\text{new}}$.

	E_c	$\tilde{E}_\text{c}^\text{d}(\tilde{v}^\text{d})$	$\tilde{E}_\text{c}^\text{d}(\tilde{u}^\text{d})$	$\tilde{E}_\text{c}^\text{s}(\tilde{v}^\text{s})$	$\tilde{E}_\text{c}^\text{s}(\tilde{u}^\text{s})$
$p_\text{a} = 1$	0.792	0.776	0.740	-	-
$p_\text{a} = 0.99$	-	-	-	0.929	0.904
$p_\text{a} = 0.5$	-	-	-	0.805	0.781

casts under-estimate required charge with $\tilde{E}_\text{c}^\text{d}(\tilde{u}^\text{d})$ being further away from E_c than $\tilde{E}_\text{c}^\text{d}(\tilde{v}^\text{d})$. In both stochastic forecasts, a full battery is not enough for $p_\text{a} = 0.99$. Furthermore, even for $p_\text{a} = 0.5$, almost all the battery energy is required and thus safe arrival cannot be guaranteed. In fact, The maximum attainability for $\tilde{E}_\text{c}^\text{s}(\tilde{v}^\text{s})$ is $q = 0.523$ and for $\tilde{E}_\text{c}^\text{s}(\tilde{u}^\text{s})$, it is $q = 0.619$. This example illustrates the importance of stochastic forecasts because the deterministic solution implies safe arrival with the respective SOC, where in fact the chance of stranding is almost 50 %. In addition, the aged battery example shows that including a battery model is vital for estimating range in an electric vehicle, as energy content and uncertainty have a strong influence on range. It can be imagined that estimating battery energy content purely based on SOC leads to a significant stranding risk. The question of which forecast model performs best in predicting attainability will be answered in Chapter 6.

Table 5.3: Quantitative results for attainability in the aged battery scenario $\hat{E}_{b,\text{aged}}$.

	E_c	$\tilde{E}_\text{c}^\text{d}(\tilde{v}^\text{d})$	$\tilde{E}_\text{c}^\text{d}(\tilde{u}^\text{d})$	$\tilde{E}_\text{c}^\text{s}(\tilde{v}^\text{s})$	$\tilde{E}_\text{c}^\text{s}(\tilde{u}^\text{s})$
$p_\text{a} = 1$	0.989	0.970	0.925	-	-
$p_\text{a} = 0.99$	-	-	-	-	-
$p_\text{a} = 0.5$	-	-	-	0.992	0.963

6 Implementation and Test

After introducing the methodology for range estimation in the previous chapters, the performance of the presented models and algorithms must be evaluated, which is the goal of this chapter. In the first part, Section 6.1, the quantitative results of the different models are presented objectively. In the second part, Section 6.2, these results are discussed and implications as well as comments are provided.

6.1 Performance Results

In this section, the algorithms for range estimation are evaluated and their performance demonstrated using the test drive dataset from the ICEV. First, the approach for evaluating the algorithms is introduced in Section 6.1.1. Subsequently, a central part of performance evaluation, the performance index used in this work, is explained in Section 6.1.2. After the evaluation method is thus introduced, the results for velocity prediction (VP), energy consumption forecast (ECF) and attainability are presented in Section 6.1.3 to Section 6.1.5.

6.1.1 Evaluation Method

To analyse the model performance of the data-driven approaches presented in this work, an evaluation method must be selected. Thereby, the *out-of-sample* performance is important, as presented in Section 2.4. In that section, the concept of cross validation was shown which is now applied on the complete data set. As illustrated in Section 3.4, the total data set comprises of 10 test drivers each with trips A and B (Figure 3.8). Consequently, 2-fold cross validation can be used for each driver, resulting in 20 different forecasts for evaluating out-of-sample performance as presented in Figure 6.1. With this evaluation method, *driver specific* performance can be analysed for all trips.

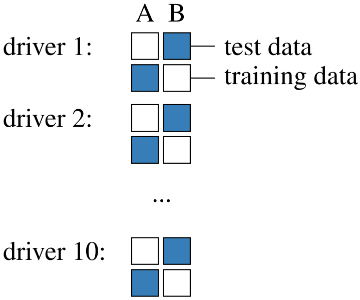


Figure 6.1: 2-fold cross-validation for analysing the results. The test drives for each of the 10 drivers have two trips A and B. Each of those is used for testing and training the models.

6.1.2 Forecast Performance Index

To evaluate the performance of stochastic forecasts, standard error measures, e.g. mean percentage error (MPE) or root mean squared error (RMSE), are not applicable since they cannot examine the stochastic aspect of the forecast. Instead of comparing two points, estimation and measurement, a predictive distribution has to be compared to the measurement. Thereby, the measurement represents a specific realization of possible events and is thus deterministic while the forecast is stochastic. Assessing the forecast quality in this case is achieved by so-called *scoring rules* [40].

Assuming the forecasted cumulative density function Z and the measurement (realization) x, then the scoring rule $S(Z,x)$ quantifies the performance in that instance. In stochastic forecasting, the scoring rule should evaluate the *sharpness* subject to *calibration* [39]. Thereby, calibration refers to the consistency of the forecast distribution and the measurements, meaning the magnitude of uncertainty is calculated correctly and neither over- nor underestimated. Sharpness refers to the concentration of the forecast distribution itself. In addition, the scoring rule should be *proper*, therefore

$$S(Z = Y, Y) \geq S(Z \neq Y, Y) \tag{6.1}$$

for the true distribution Y, meaning the obtained score is higher if the forecast distribution Z equals the true distribution Y compared to other forecast distributions.

A common proper scoring rule is the continuous ranked probability score (CRPS):

$$\mathrm{CRPS}(Z, x) = \int_{-\infty}^{\infty} \{Z(y) - \Gamma(y - x)\}^2 \, \mathrm{d}y \tag{6.2}$$

where Γ is a heaviside-function returning 1 if the argument is positive or zero, else 0. Z denotes the cumulative density function of the forecast and x the observation. In the above formulation, the CRPS is negatively oriented, hence better results achieve lower values. CRPS expresses the results in the unit of the forecast variable and is therefore easy to interpret. For deterministic forecasts, it reduces to the absolute error (AE). Thus, it offers the advantage of being able to compare stochastic and deterministic forecasts.

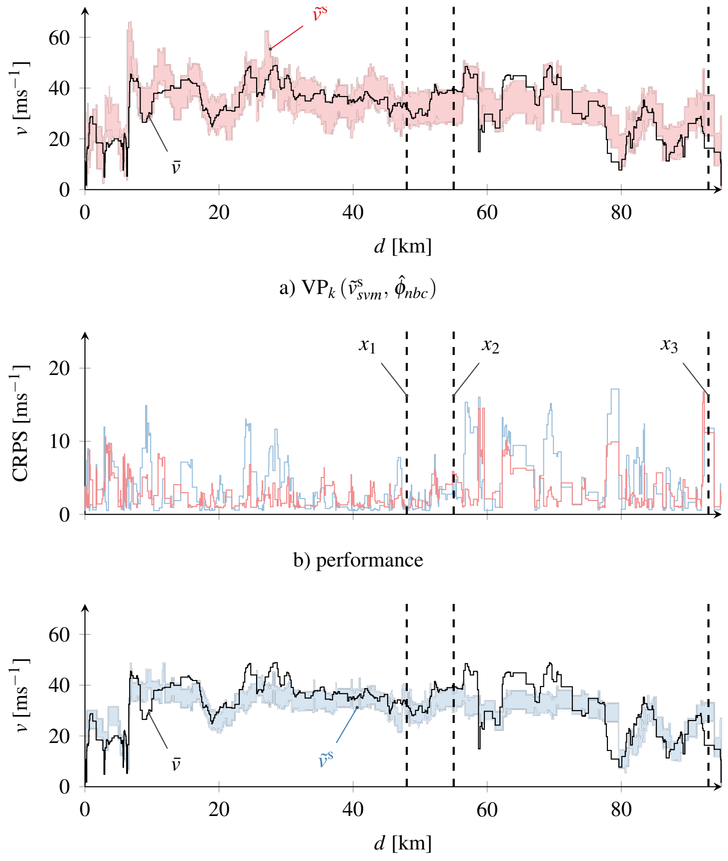
Usually, the overall forecast assessment is achieved by the mean CRPS over the individual predictions, implying equal weight of the predictions. Since in the range estimation application, predictions are executed for the individual segments k of the future trip, weights are governed by the segment lengths l_k. Therefore, it is more important to obtain good results on longer segments. Consequently, the weighted continuous ranked probability score (WCRPS) is introduced as follows:

$$\mathrm{WCRPS} = \frac{\sum_k \mathrm{CRPS}_k(Z_k, x_k) \cdot l_k}{\sum_k l_k} \,. \tag{6.3}$$

Taking a velocity prediction (VP) example, the performance evaluation with CRPS can be demonstrated. Figure 6.2a shows the measurement $\bar{v}$ and stochastic velocity prediction $\tilde{v}^{\mathrm{s}}$, where the 90 % PI is illustrated in the shaded area. Thereby, the red forecast in Figure 6.2a uses a prediction model with SVM in combination with NBC for estimating traffic phase $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{s}}_{svm}, \hat{\phi}_{nbc})$ and the blue prediction in Figure 6.2c is based on a model neglecting traffic phase $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{s}}_{svm})$. In Figure 6.2b, the corresponding CRPS values in the respective colors can be seen. When the measurement $\bar{v}$ is close to the expected value of the distribution $\tilde{v}$, a low CRPS value is achieved, as in x_1. At x_2, $\bar{v}$ is further away from the expected value but still within the PI, hence the CRPS is higher. The worst case for a forecast is when the measurement $\bar{v}$ is outside the PI, therefore very high CRPS are obtained, as in x_3. The resulting stochastic forecast in Figure 6.2c has lower uncertainty than the model including traffic phase but often the measurement is not within the PI limits. In consequence, CRPS is often higher than for $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{s}}_{svm}, \hat{\phi}_{nbc})$. This can be indicated with the overall WCRPS following (6.3), which is $3.14\,\mathrm{m\,s^{-1}}$ in the case of $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{s}}_{svm}, \hat{\phi}_{nbc})$ and $4.13\,\mathrm{m\,s^{-1}}$ for $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{s}}_{svm})$. In this example, the proper scoring rule indicates that including a traffic phase estimate will improve predictive performance. The question whether this is true for all test drives and not just the example will be answered in the following sections using the presented performance index.

6.1.3 Velocity Prediction

The velocity prediction (VP) algorithms were presented in Section 5.1. They use TRDB data and the past driving speed to predict the velocity $\tilde{v}$ on a selected test drive. The out-of-sample error is evaluated with the WCRPS over the whole dataset comprising 20 forecasts according to the cross-validation method shown in Section 6.1.1. The models for VP generate deterministic forecasts $\tilde{v}^{\mathrm{d}}$ and stochastic forecasts $\tilde{v}^{\mathrm{s}}$, each will be evaluated. One velocity prediction algorithm for long range trips is using traffic informa-



a) $\mathrm{VP}_k\left(\tilde{v}^{s}_{svm}, \hat{\phi}_{nbc}\right)$

b) performance

c) $\mathrm{VP}_k\left(\tilde{v}^{s}_{svm}\right)$

Figure 6.2: Visualization of the performance index used for the forecasts. Figure 6.2a shows the stochastic forecast VP with SVM and TPC with NBC. Figure 6.2b illustrates the continuous ranked probability score (CRPS) on each segment. At x_1, the CRPS value is low because the expected value of the predicted distribution and the measurement are close. At x_2, CRPS is higher because v is further away from the expected value. Point x_3 shows that CRPS is very high if the forecast distribution fails to cover the realization. For comparison, Figure 6.2c shows another VP with SVM but without using TPC.

tion of probes $\tilde{u}$ and was shown in Section 5.1.1. The resulting forecasts on every segment k are denoted $\mathrm{VP}_k\left(\tilde{u}^{\mathrm{d}}\right)$ for the deterministic case and $\mathrm{VP}_k\left(\tilde{u}^{\mathrm{s}}\right)$ for the stochastic case. Extending this model with learned driving behaviour can be achieved with the multi-linear regression (MLR) presented in Section 5.1.3 and the support vector machine (SVM) introduced in Section 5.1.4 which leads to the prediction models denoted $\mathrm{VP}_k\left(\tilde{v}_{mlr}\right)$ and $\mathrm{VP}_k\left(\tilde{v}_{svm}\right)$. Thereby, the stochastic and deterministic alternatives are indicated with superscripted $(\cdot)^{\mathrm{s}}$ and $(\cdot)^{\mathrm{d}}$ according to the convention used in this work. The additional feature ϕ generated by the traffic phase classification (TPC) algorithm from Section 4.2 can be included in the regression algorithms. Among them, two models are selected: traffic phase obtained by naive Bayesian classifier (NBC) $\hat{\phi}_{nbc}$, which achieved the best result, and traffic phase obtained by rule-based algorithm (RBA) $\hat{\phi}_{rba}$ which is the implementation having lowest complexity and computing time. Combining the TPC method with the regression algorithms gives four more VP algorithms: $\mathrm{VP}_k\left(\tilde{v}_{mlr}, \hat{\phi}_{rba}\right)$, $\mathrm{VP}_k\left(\tilde{v}_{mlr}, \hat{\phi}_{nbc}\right)$, $\mathrm{VP}_k\left(\tilde{v}_{svm}, \hat{\phi}_{rba}\right)$ and $\mathrm{VP}_k\left(\tilde{v}_{svm}, \hat{\phi}_{nbc}\right)$.

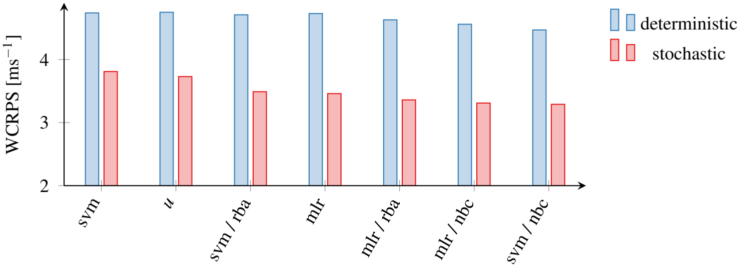


Figure 6.3: Comparison of stochastic and deterministic alternatives of the velocity prediction with the WCRPS as error measure indicating performance. The algorithms are ordered depending on their stochastic WCRPS result.

Figure 6.3 illustrates the performance of each VP algorithm in terms of WCRPS on the whole data set. The algorithms are shown in descending order depending on their stochastic WCRPS value. On first glance, it can be seen that stochastic algorithms generally perform significantly better than the deterministic alternatives, which can also be seen by the quantitative results presented in Table 6.1. The best overall result is achieved by the stochastic forecast $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{s}}_{svm},\, \hat{\phi}_{nbc})$ with WCRPS $= 3.29\,\mathrm{m\,s^{-1}}$. The worst result is the deterministic forecast using traffic information $\mathrm{VP}_k\,(\tilde{u}^{\mathrm{d}})$ with WCRPS $= 4.75\,\mathrm{m\,s^{-1}}$. Another notable fact is that the best performing deterministic forecast, $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{d}}_{svm},\, \hat{\phi}_{nbc})$ with WCRPS $= 4.47\,\mathrm{m\,s^{-1}}$ is still worse than the worst stochastic forecast $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{s}}_{svm})$ with WCRPS $= 3.81\,\mathrm{m\,s^{-1}}$. In general, the stochastic forecasts are better than the deterministic alternatives for each individual algorithms. Also, with the exception of $\mathrm{VP}_k\,(\tilde{v}_{mlr})$, the algorithms making use of the traffic phase information achieve better results. Interestingly, the SVM algorithms perform very differently, from being the worst stochastic forecast without traffic phase information $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{s}}_{svm})$ to the best overall result with $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{s}}_{svm},\, \hat{\phi}_{nbc})$. The MLR algorithms however are ranked next to each other with increasing complexity in the TPC method leading to better results, from no TPC with $\mathrm{VP}_k\,(vps_{mlr})$ over the rule-based model $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{s}}_{mlr},\, \hat{\phi}_{rba})$ to the Bayesian approach $\mathrm{VP}_k\,(\tilde{v}^{\mathrm{s}}_{mlr},\, \hat{\phi}_{nbc})$. Except for pure SVM, all regression algorithms are superior to using only probe information. The discussion of these results can be found in Section 6.1.3.

Table 6.1: Performance of the velocity prediction with the WCRPS as error measure.

	WCRPS deterministic	WCRPS stochastic
$\mathrm{VP}_k\,(\tilde{v}_{svm})$	$4.74\,\mathrm{m\,s^{-1}}$	$3.81\,\mathrm{m\,s^{-1}}$
$\mathrm{VP}_k\,(\tilde{u})$	$4.75\,\mathrm{m\,s^{-1}}$	$3.73\,\mathrm{m\,s^{-1}}$
$\mathrm{VP}_k\,(\tilde{v}_{svm},\, \hat{\phi}_{rba})$	$4.71\,\mathrm{m\,s^{-1}}$	$3.49\,\mathrm{m\,s^{-1}}$
$\mathrm{VP}_k\,(\tilde{v}_{mlr})$	$4.73\,\mathrm{m\,s^{-1}}$	$3.46\,\mathrm{m\,s^{-1}}$
$\mathrm{VP}_k\,(\tilde{v}_{mlr},\, \hat{\phi}_{rba})$	$4.63\,\mathrm{m\,s^{-1}}$	$3.36\,\mathrm{m\,s^{-1}}$
$\mathrm{VP}_k\,(\tilde{v}_{mlr},\, \hat{\phi}_{nbc})$	$4.56\,\mathrm{m\,s^{-1}}$	$3.31\,\mathrm{m\,s^{-1}}$
$\mathrm{VP}_k\,(\tilde{v}_{svm},\, \hat{\phi}_{nbc})$	$4.47\,\mathrm{m\,s^{-1}}$	$3.29\,\mathrm{m\,s^{-1}}$

6.1.4 Energy Consumption Forecast

After the performance of the VP was analysed in the previous section, the energy consumption forecast (ECF) can be evaluated with the same error measure and approach of cross-validation. Recalling the overall method from Figure 5.1, the ECF uses the VP as input information in addition to the TRDB data and the estimated vehicle parameters $\hat{\mathbf{p}}$. With these input data, a sequential Monte-Carlo simulation (MCS) calculates predicted energy consumption on each segment $\tilde{E}_{\mathrm{c},k}$, as shown in Section 5.2. Thereby, the MCS method is equivalent for all VPs, therefore the algorithms to evaluate as well as their notation are the same compared to Section 6.1.3. For example, the stochastic energy consumption forecast based on traffic information on every segment k is denoted $\mathrm{ECF}_k\left(\tilde{u}^{\mathrm{s}}\right)$ whereas the deterministic alternative is written $\mathrm{ECF}_k\left(\tilde{u}^{\mathrm{d}}\right)$. However, MCS performance depends on the number of particles ρ. To investigate the influence, MCS is performed with $\rho = 10$ and $\rho = 100$ for the total data set and the WCRPS results are shown in Figure 6.4. As in the analysis of VP, the algorithms are ranked depending on their result in descending order. Thereby, only the stochastic algorithms are shown because MCS is not performed for deterministic forecasts.

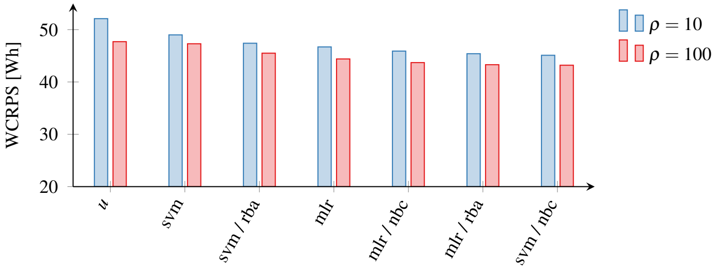


Figure 6.4: Illustration of the particle number influence on the energy consumption forecast (ECF) performance on all test drives in terms of WCRPS. The algorithms are ordered depending on their WCRPS result.

It can be seen that more particles lead to better results in every one of the considered algorithms but the relative order of the algorithms does not change. The worst performance is yielded by the forecast using probe information $\mathrm{ECF}_k\,(\tilde{u}^{\mathrm{s}})$ and the best performance is achieved by the most complex algorithm $\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{svm},\,\hat{\phi}_{nbc})$. With the exception of $\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{svm},\,\hat{\phi}_{rba})$, the algorithms using traffic phase as a feature obtain lower WCRPS values than the others. In general, the ranking of the algorithms is similar to the VP results because only $\mathrm{ECF}_k\,(\tilde{u}^{\mathrm{s}})$ and $\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{svm})$ as well as $\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{mlr},\,\hat{\phi}_{rba})$ and $\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{mlr},\,\hat{\phi}_{nbc})$ switched places.

Table 6.2: Performance of the energy consumption forecast with the sequential Monte-Carlo simulation (MCS) using $\rho = 10$.

	calculation time	WCRPS stochastic
$\mathrm{ECF}_k\,(\tilde{u}^{\mathrm{s}})$	100 %	52.1 Wh
$\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{svm})$	100 %	49.0 Wh
$\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{svm},\,\hat{\phi}_{rba})$	100 %	47.4 Wh
$\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{mlr})$	100 %	46.7 Wh
$\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{mlr},\,\hat{\phi}_{nbc})$	100 %	45.9 Wh
$\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{mlr},\,\hat{\phi}_{rba})$	100 %	45.4 Wh
$\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{svm},\,\hat{\phi}_{nbc})$	100 %	45.1 Wh

The numeric WCRPS values for $\rho = 10$ are shown in Table 6.2. Computation time is set to 100 % and serves as a base for comparing calculation demand. Table 6.3 shows the results for $\rho = 100$, where the left column shows the increase in calculation time and the right column shows the stochastic WCRPS as well as its relative deviation from the 10 particles solution. From the numeric results in Table 6.3, the increase in computation time is apparent, ranging between seven and almost eight times of the MCS with 10 particles. In contrast, the performance in terms of WCRPS improved by

4 % to 8.4 %, depending on the model. However, the overall ranking of the algorithms and thus their relative performance remains unchanged.

Table 6.3: Performance of the energy consumption forecast with the sequential Monte-Carlo simulation (MCS) using $\rho = 100$.

	calculation time	WCRPS stochastic
$\mathrm{ECF}_k\,(\tilde{u}^{\mathrm{s}})$	762 %	47.7 Wh (91.6 %)
$\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{svm})$	751 %	47.3 Wh (96.5 %)
$\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{svm},\, \hat{\phi}_{rba})$	749 %	45.5 Wh (96.0 %)
$\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{mlr})$	782 %	44.4 Wh (95.1 %)
$\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{mlr},\, \hat{\phi}_{nbc})$	704 %	43.7 Wh (95.2 %)
$\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{mlr},\, \hat{\phi}_{rba})$	743 %	43.3 Wh (95.4 %)
$\mathrm{ECF}_k\,(\tilde{v}^{\mathrm{s}}_{svm},\, \hat{\phi}_{nbc})$	755 %	43.2 Wh (95.8 %)

To get the total energy consumption forecast (ECF), the predictions of the stochastic forecasts on every segment k must be convoluted as presented in Section 5.2.3. It basically represents the sum of all segment energies $\tilde{E}^{\mathrm{s}}_{\mathrm{c},k}$ to one forecast for the total trip $\tilde{E}^{\mathrm{s}}_{\mathrm{c}}$. The results of the convolution using the introduced models and the MCS with $\rho = 10$ can be seen in Figure 6.5 and the numeric WCRPS values in Table 6.4. Although for the deterministic forecasts, neither MCS nor convolution is required, they are shown for comparison as in the VP evaluation. Thereby, the total forecast represents the aggregated value, therefore the subscript k is omitted for the algorithms. The worst performance is achieved by the deterministic traffic model $\mathrm{ECF}\,(\tilde{u}^{\mathrm{d}})$ with $\mathrm{WCRPS} = 1.88\,\mathrm{kWh}$ and the lowest error is yielded by the stochastic forecast $\mathrm{ECF}\,(\tilde{v}^{\mathrm{s}}_{mlr},\, \hat{\phi}_{nbc})$ with $\mathrm{WCRPS} = 0.63\,\mathrm{kWh}$ which is an improvement by three times. For each model, the stochastic forecast is better than the deterministic forecast. Contrary to the VP_k results, in the convoluted outcome ECF, the stochastic forecasts are not generally better than the deterministic models. For example, $\mathrm{ECF}\,(\tilde{v}^{\mathrm{d}}_{mlr},\, \hat{\phi}_{nbc})$ has a lower WCRPS than

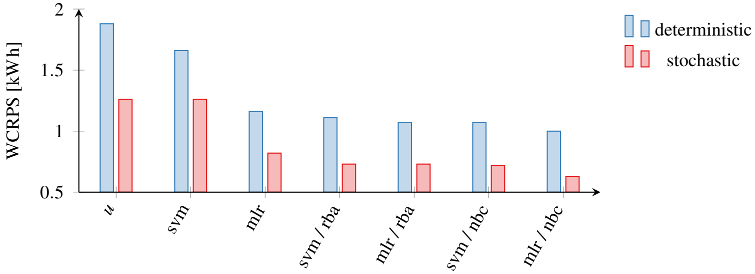


Figure 6.5: Comparison of stochastic and deterministic alternatives of the convoluted ECF with the WCRPS as error measure indicating performance. The algorithms are ordered depending on their stochastic WCRPS result.

$\mathrm{ECF}\left(\tilde{u}^{\mathrm{s}}\right)$. In addition, the ranking of the algorithms changes compared to ECF_k. For ECF, the algorithms not using traffic phase information ϕ generally perform worse than the others without exception. Better results are gained by including the RBA traffic phase model and the best models use the NBC algorithm. Also, the MLR regression improves the WCRPS value compared to using the SVM.

Table 6.4: Performance of the convoluted energy consumption forecast with the WCRPS as error measure. For the inherent sequential Monte-Carlo simulation (MCS), $\rho = 10$ is used.

	WCRPS deterministic	WCRPS stochastic
$\mathrm{ECF}\left(\tilde{u}\right)$	1.88 kWh	1.26 kWh
$\mathrm{ECF}\left(\tilde{v}_{svm}\right)$	1.66 kWh	1.26 kWh
$\mathrm{ECF}\left(\tilde{v}_{mlr}\right)$	1.16 kWh	0.82 kWh
$\mathrm{ECF}\left(\tilde{v}_{svm}, \hat{\phi}_{rba}\right)$	1.11 kWh	0.73 kWh
$\mathrm{ECF}\left(\tilde{v}_{mlr}, \hat{\phi}_{rba}\right)$	1.07 kWh	0.73 kWh
$\mathrm{ECF}\left(\tilde{v}_{svm}, \hat{\phi}_{nbc}\right)$	1.07 kWh	0.72 kWh
$\mathrm{ECF}\left(\tilde{v}_{mlr}, \hat{\phi}_{nbc}\right)$	1.00 kWh	0.63 kWh

The performance difference in using 10 or 100 particles was analysed for ECF_k and subsequently, it can be illustrated for the convoluted solution ECF. The results for $\rho = 100$ are illustrated in Table 6.5, where the absolute performance as well as the relative improvement compared to the solution with $\rho = 10$ is shown. Because the deterministic results remain unchanged, they are omitted in the table. As before, the relative order of the algorithms does not change with increasing particle number, although the relative improvement differs, ranging between 0 % and 9.8 %. The best algorithm $\mathrm{ECF}\,(\tilde{v}^{\mathrm{s}}_{mlr}, \hat{\phi}_{nbc})$ did not improve compared to using less particles and the model using only MLR without TPC benefited the most from an increase in ρ. The relative WCRPS improvements of the convoluted results are not equivalent to the WCRPS improvements on the segments ECF_k from Table 6.3 but range in the same order of magnitude. For a discussion of these results, see Section 6.2.2.

Table 6.5: Performance of the convoluted energy consumption forecast with the WCRPS as error measure. For the inherent sequential Monte-Carlo simulation (MCS), $\rho = 100$ is used.

	WCRPS stochastic
$\mathrm{ECF}\,(\tilde{u}^{\mathrm{s}})$	1.21 kWh (96.0 %)
$\mathrm{ECF}\,(\tilde{v}^{\mathrm{s}}_{svm})$	1.20 kWh (95.2 %)
$\mathrm{ECF}\,(\tilde{v}^{\mathrm{s}}_{mlr})$	0.74 kWh (90.2 %)
$\mathrm{ECF}\,(\tilde{v}^{\mathrm{s}}_{svm}, \hat{\phi}_{rba})$	0.72 kWh (98.6 %)
$\mathrm{ECF}\,(\tilde{v}^{\mathrm{s}}_{mlr}, \hat{\phi}_{rba})$	0.71 kWh (97.3 %)
$\mathrm{ECF}\,(\tilde{v}^{\mathrm{s}}_{svm}, \hat{\phi}_{nbc})$	0.70 kWh (97.2 %)
$\mathrm{ECF}\,(\tilde{v}^{\mathrm{s}}_{mlr}, \hat{\phi}_{nbc})$	0.63 kWh (100 %)

6.1.5 Attainability

The stochastic range estimation concept uses attainability p_a to express the probability of a destination being attainable, as introduced in Section 3.2. For that, the total energy consumption forecast (ECF) $\tilde{E}_\mathrm{c}$ is used, which was evaluated in the previous section. The other factor influencing attainability is the energy content estimate $\hat{E}_\mathrm{b}$ from the battery model explained in Section 3.3. Using the test drive dataset and the cross-validation from Section 6.1.1, attainability can be calculated deterministically and stochastically, following the method presented in Section 5.3. The results in terms of WCRPS with a new battery scenario and the MCS using $\rho = 10$ can be seen in Figure 6.6 and the numeric values are shown in Table 6.6. Notation and ranking order are used in accordance with the previous sections.

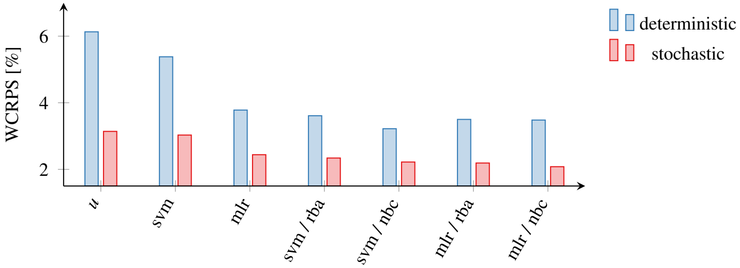


Figure 6.6: Comparison of stochastic and deterministic alternatives of the attainability prediction with the WCRPS as error measure indicating performance. The algorithms are ordered depending on their stochastic WCRPS result.

The best model for attainability prediction is the stochastic forecast $p_\mathrm{a}\left(\tilde{v}^\mathrm{s}_{mlr}, \hat{\phi}_{nbc}\right)$ with WCRPS $= 2.08\,\%$ and the worst performance comes from the deterministic traffic model $p_\mathrm{a}\left(\tilde{u}^\mathrm{d}\right)$ with WCRPS $= 6.13\,\%$. Therefore, an improvement by three times can be achieved, which is consistent with the results from ECF. For each algorithm, the stochastic forecast is better than deterministic forecast. This also holds in general as the worst stochastic result ECF $\left(\tilde{u}^\mathrm{s}\right)$ is still better than the best deterministic result $p_\mathrm{a}\left(\tilde{v}^\mathrm{s}_{svm}, \hat{\phi}_{nbc}\right)$.

The ranking of the algorithms in accordance with their stochastic result is different when evaluating attainability compared to ECF. The three models with highest WCRPS values neglect the traffic phase information. The two best models use traffic phase information together with the MLR regression approach, which achieves better results than the SVM regression.

Table 6.6: Performance of the attainability forecast with the WCRPS as error measure and the new battery model.

	WCRPS deterministic	WCRPS stochastic
$p_{\mathrm{a}}(\tilde{u})$	6.13 %	3.14 %
$p_{\mathrm{a}}(\tilde{v}_{svm})$	5.38 %	3.03 %
$p_{\mathrm{a}}(\tilde{v}_{mlr})$	3.78 %	2.44 %
$p_{\mathrm{a}}(\tilde{v}_{svm}, \hat{\phi}_{rba})$	3.61 %	2.34 %
$p_{\mathrm{a}}(\tilde{v}_{svm}, \hat{\phi}_{nbc})$	3.22 %	2.22 %
$p_{\mathrm{a}}(\tilde{v}_{mlr}, \hat{\phi}_{rba})$	3.50 %	2.19 %
$p_{\mathrm{a}}(\tilde{v}_{mlr}, \hat{\phi}_{nbc})$	3.48 %	2.08 %

Apart from evaluating performance with WCRPS, the results can be analysed more directly with help of Figure 6.7. Thereby, Figure 6.7a illustrates the overall variability of necessary SOC to complete the test drive, calculated from E_{c} and the battery model. It can be seen that due to different driving style, traffic and elevation profile, between $q_{\mathrm{real}} = 0.72$ and $q_{\mathrm{real}} = 0.97$ was required. Also, the issue of necessary *safety margin* raised in Section 3.2 can be addressed. Traditionally, deterministic forecasts are used in range estimation and a safety margin is added by the system to ensure reaching the destination. Taking the worst and best performing deterministic algorithms, the *theoretically* needed relative safety margin $\Delta b_{\mathrm{rel}}^{\mathrm{th}}$ can be calculated with

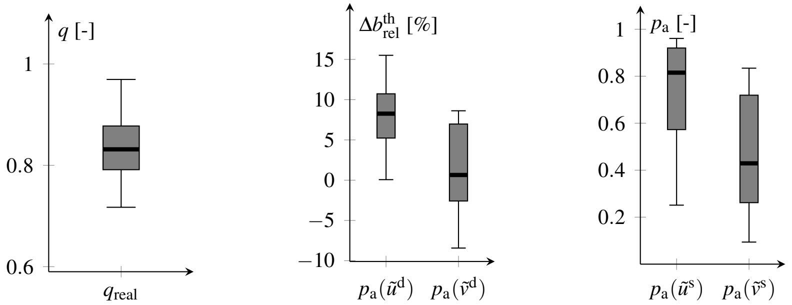
$$\Delta b_{\text{rel}}^{\text{th}} = \frac{q[p_{\text{a}} = 1] - q_{\text{real}}}{q[p_{\text{a}} = 1]} \tag{6.4}$$

where $q[p_{\text{a}} = 1]$ represents the SOC required to complete the trip according to the deterministic attainability prediction and q_{real} is the really needed SOC. In Figure 6.7b, the results for each trip and the deterministic predictions $p_{\text{a}}\left(\tilde{u}^{\text{d}}\right)$ as well as $p_{\text{a}}\left(\tilde{v}_{mlr}^{\text{d}}, \hat{\phi}_{nbc}\right)$ are shown. Using the state-of-the-art model with mean traffic speed, between 0.1 and 15.5 % more charge should have been reserved to complete the trip, the median is 8.3 %. For the model using deterministic $p_{\text{a}}\left(v_{mlr}, \phi_{nbc}\right)$, a maximum of 8.6 % more charge would have been sufficient. In some cases, the predicted energy was too high and too much energy was in the battery, as indicated by the negative Δb_{rel} values with a minimum of -8.4 %. The median is 0.6 %. From these boxplots, it can be seen that a forecast using mean traffic speed is always underestimating energy consumption in this data set thus there is always a risk of stranding. The regression-based prediction is calibrated better around the true value but in around half the cases, the battery energy would not be sufficient.

The last boxplot shows the relation between prediction and realization for the stochastic forecasts (Figure 6.7c). There, the predicted attainability $p_{\text{a}}(q_{\text{real}})$ at the really required SOC is shown for all the test drives. For the probe distribution forecast, p_{a} ranges between 25.1 % and 96.1 % with the median being 81.5 %. In contrast, the stochastic forecast model with MLR and NBC obtained p_{a} values between 10.4 % and 83.4 % with a median of 42.9 %. In comparison, an ideally calibrated forecast would achieve an expected value of 50 %. For a discussion of these results, see Section 6.2.3.

6.2 Discussion of the Presented Results

After presenting the results of the algorithms in Section 6.1, this section presents the analysis and discussion of the used forecasting models. The



a) required q for all trips

b) theroretical buffer required

c) p_{a} prediction for required q

Figure 6.7: Boxplots illustrating more details into the test drive results than WCRPS. Figure 6.7a shows the variability of the required SOC to complete the trips. If deterministic traffic prediction and deterministic MLR with NBC are used, the theoretically needed safety margin is illustrated in Figure 6.7b. For the stochastic versions, the predicted attainability can be seen in Figure 6.7c.

goal is to offer an explanation of the model performance in velocity prediction (VP), energy consumption forecast (ECF) and attainability calculation p_{a} to the reader. Insights and initial conclusions from the results are developed, and the author's opinion on the algorithms is included which makes this section more subjective than the previous ones.

6.2.1 Velocity Prediction

The velocity prediction (VP) results were shown in Section 6.1.3. Given the fact that performing VP includes human behaviour, the forecast can never be perfectly accurate and a certain degree of uncertainty persists. This hypothesis proves to be true, because for every model, the stochastic forecast achieves better results than the deterministic alternative. Hence, apart from the subjective advantage of offering the additional information of *how certain* the forecast is, including uncertainty objectively leads to better predictions. Among the deterministic algorithms, the simplest model using only mean traffic speed offers the worst result and the most complex algorithm using naive Bayesian classifier (NBC) for identifying the traffic phase and support vector machine (SVM) for driver behaviour learning shows the best performance. This is expected, as increased modelling effort should also pay off when looking at the result. Consequently, the inclusion of traffic phase identification algorithms improves the forecast. A surprising result is that using SVM does not generally mean better performance, e.g. the combination of SVM and RBA performs worse than MLR. It seems that the machine learning algorithm needs the best possible features and does not deal well with errors in the traffic phase.

Looking at the stochastic results, also the most complex model shows the best forecast, which is expected. Close behind is the prediction using MLR and the same TPC algorithm. From that, one could conclude that estimating TPC is most important, as the best two algorithms use NBC, which has been identified as the best traffic phase identification method in Section 4.2. This statement would be further supported by the fact that the two worst algorithms do not use TPC at all. However, it is not strictly true for velocity prediction (VP) since pure MLR implementation is better than the SVM in combination with RBA supplying traffic phase. This exception, however, does not blur the fact that including TPC mostly means better forecasts.

When analysing the algorithms learning driver behaviour, it is also expected that increased complexity leads to better results, meaning SVM is best, followed by MLR and pure traffic information being last. Surprisingly, it is only the case that MLR is better than traffic information but for SVM, no general statement can be made. Depending on the combination with the TPC algorithm, it is best (with NBC), worst (without TPC) or in between (with RBA). It seems that SVM is more sensible to correct TPC information than MLR and therefore it must be applied carefully. The MLR algorithms behave as expected, all of them are better than the traffic information and increased complexity in TPC leads to better results.

In addition, it is worth noting that the state-of-the-art method, deterministic traffic speed prediction, shows the poorest result. Thus every algorithm presented in this dissertation would be an improvement. Also, the worst stochastic prediction is still better than the best deterministic algorithm. In consequence, adding a measure of uncertainty in the forecast is more reasonable than applying more complex algorithms and predicting deterministically. This is one of the most important insights of the velocity prediction (VP) evaluation. Another insight is that including a traffic phase estimate mostly improves forecast accuracy.

6.2.2 Energy Consumption Forecast

The results of the energy consumption forecast (ECF) algorithms were shown in Section 6.1.4. Since ECF uses the resulting forecasts from VP to calculate energy consumption, it could be assumed that the order of the forecast models follows the same ranking as for the VP presented in Section 6.1.3. However, the results show that while overall the models perform similar, the ranking differs for some algorithms. This can be explained by the non-linear dependency of energy consumption on velocity according to the powertrain model introduced in Section 4.1. By the driving resistance equations, it is quadratic already and the characteristic maps of the powertrain components are also non-linear. That means, an algorithm performing well in velocity prediction is not necessarily the best choice for energy consumption forecast. Another difference is the uncertainty propagation problem solved with MCS, which implies another calculation model that has an influence on the outcome. Following that, a forecast algorithm should always be evaluated on the exact variable to predict and not a (closely) related one.

From the evaluations of segment specific forecast ECF_k, it can be seen that increased particle number improves sampling accuracy and thus also forecast accuracy at the cost of computation time. This is a natural outcome

when using MCS but the interesting factor is the relation of improving accuracy versus computation cost. Applied to the dataset used in this work, increasing the particle number from $\rho = 10$ to $\rho = 100$ leads to a performance improvement of lower than 10 % while increasing computation time over 700 %. From this, it can be concluded that a high number of particles is only viable in an environment with a lot of computational resources, which is not the case in a control box of a modern vehicle. Therefore, in the range estimation application for online-vehicle use, the 10 particle's solution is sufficiently accurate. Still, the algorithms could be implemented in a backend connected to the vehicle, where computational resources are not such a big issue. Most importantly, the relative ranking of the algorithms does not change with increased particle number, hence their comparison can be made independently. This analysis does also hold for the convoluted results.
As in the VP, the best performance for ECF_k is achieved with the most complex solution, the stochastic forecast using support vector machine (SVM) for driver behaviour and naive Bayesian classifier (NBC) for estimating traffic phase. However, this is an intermediate result as the final solution of ECF is obtained after convolution. In that convoluted result, $\mathrm{ECF}\,(\tilde{v}^{\mathrm{s}}_{mlr}, \hat{\phi}_{nbc})$ turns out to be the best option. From the ranking, it seems as the TPC is the decisive parameter and MLR is better than SVM. Interestingly, the best algorithm, $\mathrm{ECF}\,(\tilde{v}^{\mathrm{s}}_{mlr}, \hat{\phi}_{nbc})$ does not improve with increased particle number. Theoretically, this could mean the forecast is optimal but such a hypothesis cannot be proved within this work. In summary, for the ECF, stochastic algorithms also perform better than deterministic ones and traffic phase information improves forecast quality. Surprisingly, the linear model is superior to the support vector machine (SVM) and this remains unchanged for increased particle number.

6.2.3 Attainability

In this work, attainability is used to express stochastic range estimation and therefore represents the central and most important value to be analysed. Again, it might be assumed that the forecast algorithms should be ranked in the same order as the ECF but because of the comparison with the battery model and its uncertainty, this assumption must be verified. From the results in Table 6.6, the assumption must be rejected. Whereas for ECF, the classification of traffic phase was the defining factor, when calculating attainability, it is the regression algorithm of the VP. The overall best and worst algorithms however stay the same.

From this end result, it can be concluded that the best way to improve the forecasts is the introduction of an uncertainty measure, therefore making them stochastic. In fact, even using the probe distribution without learning driver behaviour makes the range estimation already better than the most complex deterministic model used in this work. Another benefit of a stochastic forecast is the ability to assess how likely a destination is within reach. In standard applications with deterministic forecasts, either high safety margins must be chosen or the stranding of vehicles is risked. In addition, the presented concept of learning driver behaviour with respect to traffic phase leads to significantly better results than using only traffic information, as situation and driver specific knowledge is included. Even for deterministic forecasts, the required safety margins could be reduced.

Lastly, the *data-driven* model development concept is strong when clear physical dependencies are absent, in this case the behaviour of driver and traffic. Instead of trying to model the details, learning algorithms can imply the connections from data and the researcher simply gives the right boundary conditions. For example, instead of using a SVM to learn driving behaviour, in this dissertation the traffic phase is learned explicitly from relevant input signals and then used as a feature because on a high level, it should give better results. Verifying this with cross-validation gives important insights again back into model development.

7 Concluding Remarks

In this chapter, the content of the presented dissertation is summarized (Section 7.1). Subsequently, the implication of the findings on the related research is given in Section 7.2 before providing an outlook with open problems and opportunities for further research in the area (Section 7.3).

7.1 Summary

Drivers of electric vehicles suffer from range anxiety, the fear of stranding with an empty battery due to limited range and charging infrastructure. To reduce this effect, the problem of reliable range estimation for electric vehicles was analysed in this dissertation. Because various physical influences such as powertrain efficiency or vehicle mass with their measurement errors as well as non-deterministic influences such as driver behaviour and traffic state effect driving range, its estimation is subject to uncertainty. Addressing the issue of uncertainty, a stochastic expression of driving range was used, which is called attainability. Attainability represents the probability of reaching a selected destination and depends on battery energy content as well as energy consumption on the trip. While battery energy content can be modelled physically, energy consumption on the trip is influenced by driver and traffic, therefore it is non-deterministic. Forecasting energy consumption reliably was the focus of this work.

For the physical influences in energy consumption, a powertrain model of a real electric vehicle was generated and the necessary driving resistance parameters were estimated using a robust Kalman filter (Section 4.1). It was shown that using a prediction model based on the parameter estimates, the powertrain model and the kinetic energy from the velocity profile, future consumption can be calculated accurately. However, in a forecast scenario, the future velocity profile is not available. Therefore, driver behaviour with respect to the specific traffic phase must be learned during vehicle operation and this knowledge can be used to generate velocity predictions.

Hence, multiple traffic phase classification algorithms were created using graphical models as well as decision rules (Section 4.2). These approaches use signals from the CAN-bus, e.g. identified vehicles from the camera, to infer the traffic phase. In a comparison of the classification methods, the naive Bayesian classifier (NBC) achieved the best result. The information about the current phase is provided to the velocity prediction (VP) algorithm, which uses it as a feature among others to perform regression on the driving data. Two regression algorithms were implemented, a multi-linear regression (MLR) and a support vector machine (SVM) which generate predictive velocity distributions from the input data (Section 5.1).

To calculate future energy consumption from the predictive velocity distribution, a sequential Monte-Carlo simulation (MCS) with importance sampling was used (Section 5.2). The sampling distribution was obtained with a Bayesian model to obtain realistic particle behaviour. Convolution of the resulting uncertainty distribution of energy consumption on all parts of the trip gives the total energy consumption forecast (ECF). This information could be ultimately used to calculate attainability and thus, a stochastic range estimation.

A series of test drives with various drivers provided a data set where the developed algorithms could be compared. Using a proper scoring rule, deterministic and stochastic versions of forecasts were examined. It was found out that the state-of-the-art solution, where a prediction is based on mean traffic speed, offered the worst performance among the developed algorithms. Furthermore, it was shown that more complex models including driving behaviour and traffic phase improve the prediction result. Lastly, it was illustrated that stochastic forecasts generally obtain better results with more information than deterministic ones. Even simple stochastic models outperform complex deterministic solutions, which was one of the main findings in this work. Another important outcome is the benefit of the traffic phase feature in driving behaviour learning. The mean traffic speed is not sufficient to describe the traffic situation, especially for the local surrounding of the vehicle.

7.2 Contribution

This dissertation contributes several findings and verifications to the research community. Firstly, the need for stochastic forecasts in range estimation is presented and detailed analyses as well as instructions on how to express and calculate driving range stochastically are given. In related research, using deterministic predictions is the normal case and often, the uncertainty within the system is not addressed. More often, the focus lies on incorporating ever more features and calculate energy consumption with ever more complex models. The assumption that a deterministic forecast can be 100 % accurate with enough effort is a weak one and with the involvement of human interaction and behaviour, it is hardly possible. The results of this dissertation show that for a more reliable forecast, it is more helpful to include a measure of uncertainty than making the models more complex.

In this dissertation, a data-driven approach to modelling is chosen. Standard engineering practice would be to create a physical model and then compare it with measurements in an open-loop scenario. If the model is not accurate enough, more parameters are introduced or the accuracy of the chosen parameters is increased. In the data-driven approach, the physical world is only modelled to a certain extent and the details are learned using data. Using this methodology, accurate and flexible models can be created with less effort. A good example for this is the use of parameter estimation in the powertrain model or the dependency learning of graphical models in traffic phase classification. The later algorithm poses one of the main contributions this dissertation provides.

The concept of including driver behaviour via machine learning based on driving and traffic data is not new. However, to calculate the traffic situation and classify the phase explicitly from the available observations has not been introduced before to the best of my knowledge. This methodology can be used for researchers even outside the field of range estimation, because it offers a way of learning driver behaviour specific to the surrounding. The use of graphical models furthermore is interpretable to the researcher in a way that black-box approaches do not permit. This fact makes them more applicable to modern day control boxes because validation is easier.

In general, the validation of generated models is a key in current research. In this dissertation, special attention is given to the used performance indices and thus model validation capabilities. One example is the use of robust error measures for parameter estimation or the proper scoring rules for stochastic forecasts. Too often, error measures are chosen at random by researchers which could induce bias to the results.

Lastly, in related research there are many functionalities requiring reliable driving range or energy consumption estimates, such as eco-routing or charge planning. Often, simplified deterministic approaches are used but this could result in wrong decisions. For example in a charge planning scenario, the information of how likely the driver will make it to the charging station is necessary to choose which is the best option. This information could be provided by the attainability value, as shown in this work.

7.3 Outlook

In this dissertation, various algorithms and models were presented and their performance was analysed to compare them. However, in a vehicle application, not only accuracy but also computation time is important. The best model might not be the the one with the best result but the one with lowest computational demand of the control box while still achieving moderate results. While this is the case, the increased connectivity of modern vehicles and the availability of huge back-end structures nowadays offers new ways for software development. Depending on the data required, the signals and sensors used as well as the calculation complexity, algorithms can be split between vehicle control boxes and back-end. For example, the traffic phase classification which needs vehicle sensors could be installed on the control box and then send the information to the back-end where the sequential Monte-Carlo simulation (MCS) is performed which is computationally intensive. This is especially interesting since range estimation algorithms have low demands concerning real-time capability.

In addition, the research in predictive traffic phase classification should be intensified as it would increase data quality of the traffic database. However, that has to be executed by traffic scientists with access to the direct probe data instead of automotive engineers. Using more information from the vehicle sensors on the other hand could also help traffic data providers, for example since connected vehicles have position information about non-connected vehicles via the camera. Improving traffic information in such a way would not only help range estimation but also routing applications in navigation systems.

For a real-time solution, all used algorithms must be implemented in a recursive way. Also, a more detailed battery model would allow the consideration of power demand, ambient temperature etc. Furthermore, the algorithms should be validated with a lager data-set with trips worldwide. Drivers, traffic and environmental conditions change depending on the country, region or legal restriction. Finally, the concept of reliable range estimation enables optimization of charge planning scenarios. This is an important task, as in the future more electric vehicles will be on the streets.

Bibliography

[1] ALBERS, S. ; KLAPPER, D. ; KONRADT, U. ; WALTER, A. ; WOLF, J.: *Methodik der empirischen Forschung*. 3. Wiesbaden, Germany : Springer, 2009. – ISBN 978–3–8349–1702–4

[2] ALTMANNSHOFER, S. ; ENDISCH, C.: Robust vehicle mass and driving resistance estimation. In: *2016 American Control Conference (ACC)*. Boston, MA, USA : IEEE, 2016, S. 6869–6874

[3] AMBROSIUS, W.: *Methods in Molecular Biology*. Bd. 404: *Topics in Biostatistics*. 1. Totowa, NJ, USA : Humana Press Inc., 2008. – ISBN 978–1–58829–531–6

[4] ANDERSON, T. ; MATTSON, C.: Propagating Skewness and Kurtosis Through Engineering Models for Low-Cost, Meaningful, Nondeterministic Design. In: *Journal of Mechanical Design* 134 (2012), Nr. 10

[5] ANTONIOU, C. ; KOUTSOPOULOS, H. ; YANNIS, G.: Dynamic data-driven local traffic state estimation and prediction. In: *Transportation Research Part C: Emerging Technologies* 34 (2013), S. 89–107

[6] ARMSTRONG, S.: *Principles of Forecasting: A Handbook for Researchers and Practitioners*. New York, USA : Springer, 2002. – ISBN 0–7923–7930–6

[7] ASHBROOK, D. ; STARNER, T.: Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users. In: *Personal and Ubiquitous Computing* 7 (2003), Nr. 5, S. 275–286

[8] BASLER, A.: *Eine modulare Funktionsarchitektur zur Umsetzung einer gesamtheitlichen Betriebsstrategie für Elektrofahrzeuge*. Karlsruhe, Karlsruher Institut für Technologie, Dissertation, 2015

[9] BASSO, R. ; LINDROTH, P. ; KULCSAR, B. ; EGARDT, B.: Traffic aware electric vehicle routing. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Rio de Janeiro, Brazil : IEEE, 2016, S. 416–421

[10] BENDER, S. ; CHODURA, H. ; GROSS, M. ; KÜHN, T. ; WATTEROTH, V.: e-generation Research project with a positive balance. In: *Porsche Engineering Magazine* (2015), Nr. 2, S. 23–27

[11] BERECIBAR, M. ; GANDIAGA, I. ; VILLARREAL, I. ; OMAR, N. ; VAN MIERLO, J. ; VAN DEN BOSSCHE, P.: Critical review of state of health estimation methods of Li-ion batteries for real applications. In: *Renewable and Sustainable Energy Reviews* 56 (2016), S. 572–587

[12] BISHOP, C.: *Pattern Recognition and Machine Learning*. New York, USA : Springer, 2006 (Information Science and Statistics). – ISBN 978–0387–31073–2

[13] BOX, G. ; JENKINS, G. ; REINSEL, G.: *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA : John Wiley & Sons, 2015. – ISBN 978–1–118–67502–1

[14] BOZORGI, A. ; FARASAT, M. ; MAHMOUD, A.: A Time and Energy Efficient Routing Algorithm for Electric Vehicles based on Historical Driving Data. In: *IEEE Transactions on Intelligent Vehicles* 2 (2017), Nr. 4, S. 308–320

[15] CAPPE, O. ; GODSILL, S. ; MOULINES, E.: An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. In: *Proceedings of the IEEE* 95 (2007), Nr. 5

[16] CHARKHGARD, M. ; FARROKHI, M.: State-of-Charge Estimation for Lithium-Ion Batteries Using Neural Networks and EKF. In: *IEEE Transactions on Industrial Electronics* 57 (2010), Nr. 12, S. 4178–4187

[17] CHENG, Z. ; CHOW, M. ; JUNG, D. ; JEON, J.: A big data based deep learning approach for vehicle speed prediction. In: *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*. Edinburgh, UK : IEEE, 2017, S. 389–394

[18] CONWAY, T.: On the Effects of a Routing and Reservation System on the Electric Vehicle Public Charging Network. In: *IEEE Transactions on Intelligent Transportation Systems* 18 (2017), Nr. 9, S. 2311–2318

[19] CREUTZIG, F. ; JOCHEM, P. ; EDELENBOSCH, O. ; MATTAUCH , L. ; VAN VUUREN, D. ; MCCOLLUM, D. ; MINX, J.: Transport: A roadblock to climate change mitigation? Urban mobility solutions foster climate mitigation. In: *Science* 350 (2015), Nr. 6263, S. 911–912

[20] DARWICHE, Adnan: *Modeling and reasoning with Bayesian networks*. Cambridge, UK : Cambridge University Press, 2009. – ISBN 978–1–1076–7842–2

[21] DEARDOFF, Matthew D. ; WIESNER, Brady N. ; FAZIO, Joseph: Estimating Free-flow Speed from Posted Speed Limit Signs. In: *Procedia - Social and Behavioral Sciences* 16 (2011), S. 306–316

[22] DEHAY, D. ; LESKOW, J. ; NAPOLITANO, A.: Central Limit Theorem in the Functional Approach. In: *IEEE Transactions on Signal Processing* 61 (2013), Nr. 16, S. 4025–4037

[23] D'HOOGE, A. ; REBBECK, L. ; PALIN, R. ; MURPHY, Q.: Application of Real-World Wind Conditions for Assessing Aerodynamic Drag for On-Road Range Prediction. In: *SAE Technical Paper 2015-01-1551* (2015)

[24] Di Cairano, S. ; Bernardini, D. ; Bemporad, A. ; Kolmanovsky, I.: Stochastic MPC With Learning for Driver-Predictive Vehicle Control and its Application to HEV Energy Management. In: *IEEE Transactions on Control Systems Technology* 22 (2014), Nr. 3, S. 1018–1031

[25] Economist, The: *China moves towards banning the internal combustion engine*. `https://www.economist.com/business/2017/09/14/china-moves-towards-banning-the-internal-combustion-engine`. Version: 2017. – accessed 22.04.2021

[26] Eisel, M. ; Nastjuk, I. ; Kolbe, L.: Understanding the influence of in-vehicle information systems on range stress – Insights from an electric vehicle field experiment. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 43 (2016), S. 199–211

[27] Enthaler, A. ; Gauterin, F.: Method for reducing uncertainties of predictive range estimation algorithms in electric vehicles. In: *2015 IEEE 82nd Vehicular Technology Conference (VTC)*. Boston, MA, USA : IEEE, 2015, S. 1–5

[28] Enthaler, A. ; Weustenfeld, T. ; Gauterin, F. ; Köhler, J.: Thermal management consumption and its effect on remaining range estimation of electric vehicles. In: *2014 International Conference on Connected Vehicles and Expo (ICCVE)*. Vienna, Austria : IEEE, 2014, S. 170–177

[29] Ericsson, E.: Independent driving pattern factors and their influence on fuel-use and exhaust emission factors. In: *Transportation Research Part D: Transport and Environment* 6 (2001), Nr. 5, S. 325–345

[30] Fechtner, H. ; Teschner, T. ; Schmuelling, B.: Range Prediction for Electric Vehicles: Real-Time Payload Detection by Tire

Pressure Monitoring. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. Seoul, Korea (South) : IEEE, 2015, S. 767–772

[31] FENG, W. ; FIGLIOZZI, M.: An economic and technological analysis of the key factors affecting the competitiveness of electric commercial vehicles: A case study from the USA market. In: *Transportation Research Part C: Emerging Technologies* 26 (2013), S. 135–145

[32] FOLLECO, A. ; KHOSHGOFTAAR, T. ; VAN HULSE, J. ; BULLARD, L.: Identifying learners robust to low quality data. In: *2008 IEEE International Conference on Information Reuse and Integration*. Las Vegas, NV, USA : IEEE, 2008, S. 190–195

[33] FRANKE, T. ; NEUMANN, I. ; BÜHLER, F. ; COCRON, P. ; KREMS, J.: Experiencing Range in an Electric Vehicle - Understanding Psychological Barriers. In: *Applied Psychology: An International Review* 61 (2012), Nr. 3, S. 368–391

[34] FRENAY, B. ; KABAN, A.: A Comprehensive Introduction to Label Noise. In: *Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. Bruges, Belgium : i6doc.com.publ, 2014, S. 667–676

[35] FRIEDMANN, N. ; GEIGER, D. ; GOLDSZMIDT, M.: Bayesian Network Classifiers. In: *Machine Learning* 29 (1997), S. 131–163

[36] FÜNFGELD, S. ; HOLZÄPFEL, M. ; FREY, M. ; GAUTERIN, F.: Stochastic Forecasting of Vehicle Dynamics Using Sequential Monte Carlo Simulation. In: *IEEE Transactions on Intelligent Vehicles* 2 (2017), Nr. 2, S. 111–122

[37] GELMAN, A.: Analysis of variance—why it is more important than ever. In: *The Annals of Statistics* 33 (2005), Nr. 1, S. 1–53

[38] GEROLIMINIS, Nikolas ; DAGANZO, Carlos F.: Existence of urban-scale macroscopic fundamental diagrams. In: *Transportation Research Part B: Methodological* 42 (2008), Nr. 9, S. 759–770

[39] GNEITING, T. ; BALABDAOUI, F. ; RAFTERY, A.: Probabilistic forecasts, calibration and sharpness. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69 (2007), Nr. 2, S. 243–268

[40] GNEITING, T. ; RAFTERY, A.: Strictly Proper Scoring Rules, Prediction, and Estimation. In: *Journal of the American Statistical Association* 102 (2007), Nr. 477, S. 359–378

[41] GOOGLE INC: *What is the Google Maps API?* `https://developers.google.com/maps/`. – accessed 22.04.2021

[42] GRUBWINKLER, S. ; LIENKAMP, M.: *Advances in Intelligent Systems and Computing*. Bd. 323: *Energy Prediction for EVs Using Support Vector Regression Methods*. Cham, Switzerland : Springer, 2015. – ISBN 978–3–319–11310–4

[43] GUYON, I. ; ELISSEEFF, A.: An Introduction to Variable and Feature Selection. In: *Journal of Machine Learning Research* 3 (2003), S. 1157–1182

[44] HAKEN, K.: *Grundlagen der Kraftfahrzeugtechnik*. München, Germany : Hanser, 2008 (Fahrzeugtechnik). – ISBN 978–3–446–22812–2

[45] HAMACHER, M. ; PAPAJEWSKI, J. ; GEBHARD, P. ; RIEDEL, A. ; BENDA, V. ; FAHRENKROG, F. ; UHRNER, T. ; TÖPLER, F.: Audi Energy Assist - Efficiency and Adaptive Controlled Consumer Experience for Plug-in Hybrid Electric Vehicles. In: *22nd Aachen Colloquium Automobile and Engine Technology 2013*. Aachen, Germany : Aachen University Press, 2013, S. 1443–1464

[46] HAN, J. ; SCIARRETTA, A. ; OJEDA, L. ; NUNZIO, G. de ; THIBAULT, L.: Safe- and eco-driving control for connected and automated electric vehicles using analytical state-constrained optimal

solution. In: *IEEE Transactions on Intelligent Vehicles* 3 (2018), Nr. 2

[47] HASTIE, T. ; TIBSHIRANI, R. ; FRIEDMAN, J.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. New York, USA : Springer, 2009 (Springer Series in Statistics). – ISBN 978–0–387–84858–7

[48] HERE GLOBAL B.V.: *Traffic API Developer's Guide*. `https://developer.here.com/documentation/traffic/dev_guide/topics/quick-start.html`. – accessed 22.04.2021

[49] HOGG, R. ; MCKEAN, J. ; CRAIG, A.: *Introduction to mathematical statistics*. 6. Upper Saddle River, NJ, USA : Pearson Education International, 2005. – ISBN 0–13–122605–3

[50] HORREIN, L. ; BOUSCAYROL, A. ; LHOMME, W. ; DÉPATURE, C.: Impact of heating system on the range of an electric vehicle. In: *IEEE Transactions on Vehicular Technology* 66 (2016), Nr. 6, S. 4668 – 4677

[51] HUBER, P. J.: Robust Estimation of a Location Parameter. In: *The Annals of Mathematical Statistics* 35 (1964), Nr. 1, S. 73–101

[52] HUBER, P. J. ; RONCHETTI, E. M.: *Robust statistics*. 2. Hoboken, NJ, USA : John Wiley & Sons, 2009 (Wiley series in probability and mathematical statistics). – ISBN 9780470129906

[53] HYNDMAN, R. ; ATHANASOPOULOS, G.: *Forecasting: principles and practice*. Melbourne, Australia : OTexts, 2013 `http://otexts.org/fpp/`. – ISBN 0987507109. – accessed 22.04.2021

[54] INTERNATIONAL ENERGY AGENCY: *Global EV Outlook 2020: Entering the decade of electric drive?* `https://www.iea.org/reports/global-ev-outlook-2020`. Version: 2020. – accessed 22.04.2021

[55] JAZAR, R.: *Vehicle Dynamics: Theory and Application*. New York, USA : Springer, 2014. – ISBN 978–1–4614–8543–8

[56] JEONG, Y. ; CHO, Y. ; AHN, J. ; RYU, S. ; LEE, B.: Enhanced Coulomb counting method with adaptive SOC reset time for estimating OCV. In: *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*. Pittsburgh, PA, USA : IEEE, 2014, S. 1313–1318

[57] JIANG, B. ; FEI, Y.: Vehicle Speed Prediction by Two-Level Data Driven Models in Vehicular Networks. In: *IEEE Transactions on Intelligent Transportation Systems* 18 (2016), Nr. 7, S. 1793–1801

[58] JORDAN, M.: *Learning in Graphical Models*. Dordrecht, Netherlands : Springer, 2012. – ISBN 978–94–010–6104–9

[59] JULIER, S.: A Skewed Approach to Filtering. In: *SPIE 3373 Signal and Data Processing of Small Targets*. Orlando, FL, USA : SPIE, 1998, S. 271–282

[60] JULIER, S. ; UHLMANN, J.: A general method for approximating nonlinear transformations of probability distributions. In: *Technical report , Robotics Research Group, Department of Engineering Science, University of Oxford* (1996)

[61] JULIER, S. ; UHLMANN, J.: A Consistent, Debiased Method for Converting Between Polar and Cartesian Coordinate Systems. In: *Acquisition, Tracking, and Pointing XI*. Orlando, FL, USA : SPIE, 1997, S. 110–121

[62] JULIER, S. ; UHLMANN, J. ; DURRANT-WHYTE, H.: A new approach for filtering nonlinear systems. In: *1995 American Control Conference - ACC'95*. Seattle, WA, USA : IEEE, 1995, S. 1628–1632

[63] JUNG, M. ; STEINERT, M.: Displayed Uncertainty Improves Driving Experience and Behavior: The Case of Range Anxiety in an Elec-

tric Car. In: *33rd Annual ACM Conference on Human Factors in Computing Systems*. Seoul, Korea : ACM, 2015, S. 2201–2210

[64] KALMAN, R.: A New Approach to Linear Filtering and Prediction Problems. In: *Journal of Basic Engineering* 82 (1960), Nr. 1, S. 35–45

[65] KARBOWSKI, D. ; PAGERIT, S. ; CALKINS, A.: Energy Consumption Prediction of a Vehicle along a User-Specified Real-World Trip. In: *World Electric Vehicle Journal* 5 (2012), Nr. 4, S. 1109–1120

[66] KERNER, B.: Synchronized Flow as a New Traffic Phase and Related Problems for Traffic Flow Modelling. In: *Mathematical and Computer Modelling* 35 (2002), S. 481–508

[67] KERNER, B.: *The Physics of Traffic: empirical freeway pattern features, engineering applications, and theory*. Berlin, Germany : Springer, 2004. – ISBN 978–3–642–05850–9

[68] KERNER, B. ; DEMIR, C. ; HERRTWICH, R. ; KLENOV, S. ; REHBORN, H. ; ALEKSIC, M. ; HAUG, A.: Traffic state detection with floating car data in road networks. In: *2005 IEEE Intelligent Transportation Systems*. Vienna, Austria : IEEE, 2005, S. 44–49

[69] KUBICKA, M. ; SCIARRETTA, A. ; CELA, A. ; MOUNIER, H. ; THIBAULT, L. ; NICULESCU, S.: About prediction of vehicle energy consumption for eco-routing. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Rio de Janeiro, Brazil : IEEE, 2016, S. 1096–1101

[70] KULKARNI, V. ; MORO, A. ; GARBINATO, B.: A mobility prediction system leveraging realtime location data streams. In: *22nd Annual International Conference on Mobile Computing and Networking*. New York, USA : ACM, 2016, S. 430–432

[71] LEFÈVRE, S. ; SUN, C. ; BAJCSY, R. ; LAUGIER, C.: Comparison of Parametric and Non-Parametric Approaches for Vehicle Speed Prediction. In: *American Control Conference*. Portland, OR, USA : IEEE, 2017, S. 3494–3499

[72] LIN, J. ; ZHOU, W. ; WOLFSON, O.: Electric Vehicle Routing Problem. In: *Transportation Research Procedia* 12 (2016), S. 508–521

[73] LIU, L. ; WANG, L. ; CHEN, Z. ; WANG, C.: Integrated System Identification and State-of-Charge Estimation of Battery Systems. In: *IEEE Transactions on Energy Conversion* 28 (2012), Nr. 1, S. 12–23

[74] LU, L. ; HAN, X. ; LI, J. ; HUA, J. ; OUYANG, M.: A review on the key issues for lithium-ion battery management in electric vehicles. In: *Journal of Power Sources* 226 (2013), Nr. 1, S. 272–288

[75] LUTSEY, N. ; THE INTERNATIONAL COUNCIL ON CLEAN TRANSPORTATION (Hrsg.): *Global climate change mitigation potential from a transition to electric vehicles.* `https://theicct.org/publications/global-climate-change-mitigation-potential-transition-electric-vehicles`. Version: 2015. – accessed 22.04.2021

[76] MAFFEI, L. ; MASULLO, M.: Electric Vehicles and Urban Noise Control Policies. In: *Archives of Acoustics* 39 (2014), Nr. 3, S. 333–341

[77] MASIKOS, M. ; DEMESTICHAS, K. ; ADAMOPOULOU, E. ; THEOLOGOU, M.: Mesoscopic forecasting of vehicular consumption using neural networks. In: *Soft Computing* 19 (2015), Nr. 1, S. 145–156

[78] MILLER, R.: *Beyond ANOVA: Basics of Applied Statistics*. New York, USA : Chapman & Hall, 1997. – ISBN 978–0412070112

[79] MORGAN, M. ; HENRION, M.: *Uncertainty: a guide to dealing with uncertainty in quantitative risk and policy analysis*. Cambridge, UK : Cambridge University Press, 1990. – ISBN 0–521–36542–2

[80] MUKHERJEE, D.: Effect of Pavement Conditions on Rolling Resistance. In: *American Journal of Engineering Research (AJER)* 3 (2014), Nr. 7, S. 141–148

[81] MURPHY, K.: *Dynamic Bayesian Networks: Representation, Inference and Learning*, University of California, Berkeley, Dissertation, 2002

[82] NHA, V. ; DJAHEL, S. ; MURPHY, J.: A comparative study of vehicles' routing algorithms for route planning in smart cities. In: *2012 First International Workshop on Vehicular Traffic Management for Smart Cities (VTM)*. Dublin, Ireland : IEEE, 2012, S. 1–6

[83] NUNZIO, G. de ; THIBAULT, L. ; SCIARRETTA, A.: A model-based eco-routing strategy for electric vehicles in large urban networks. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Rio de Janeiro, Brazil : IEEE, 2016, S. 2301–2306

[84] OLIVA, J. ; WEIHRAUCH, C. ; BERTRAM, T.: A Model-Based Approach for Predicting the Remaining Driving Range in Electric Vehicles. In: *Annual Conference of the Prognostics and Health Management Society 2013*. New Orleans, USA : PHM Society, 2013, S. 1–11

[85] OMAR, N. ; VERBRUGGE B. ; MULDER G. ; VAN DEN BOSSCHE, P. ; PAUWELS, S.: Evaluation of performance characteristics of various lithium-ion batteries for use in BEV application. In: *2010 IEEE Vehicle Power and Propulsion Conference*. Lille, France : IEEE, 2010, S. 1–6

[86] ONDRUSKA, P. ; POSNER, I.: Probabilistic attainability maps: Efficiently predicting driver-specific electric vehicle range. In: *2014 IEEE Intelligent Vehicles Symposium*. Dearborn, MI, USA : IEEE, 2014, S. 1169–1174

[87] Onori, S. ; Serrao, L. ; Rizzoni, G.: *Hybrid Electric Vehicles: Energy Management Strategies*. Cham, Switzerland : Spinger, 2016 (SpringerBriefs in Electrical and Computer Engineering). – ISBN 9781447167792

[88] Owen, A.: *Monte Carlo theory, methods and examples*. `https://statweb.stanford.edu/~owen/mc/`. Version: 2013. – accessed 22.04.2021

[89] Panov, V.: *Modern Problems of Stochastic Analysis and Statistics*. Cham, Switzerland : Springer, 2017 (Springer Proceedings in Mathematics & Statistics). – ISBN 978–3–319–65312–9

[90] Papageorgiou, M. ; Diakaki, C. ; Dinopoulou, V. ; Kotsialos, A. ; Wang, Y.: Review of road traffic control strategies. In: *Proceedings of the IEEE* 91 (2003), Nr. 12, S. 2043–2067

[91] Park, J. ; Li, D. ; Murphey, Y. ; Kristinsson, J. ; McGee, R. ; Kuang, M. ; Phillips, T.: Real time vehicle speed prediction using a Neural Network Traffic Model. In: *The 2011 International Joint Conference on Neural Networks*. San Jose, CA, USA : IEEE, 2011, S. 2991–2996

[92] Pearson, K.: On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. In: *Philosophical Magazine* 50 (1900), Nr. 5, S. 157–175

[93] Picard, A. ; Fang, H.: Methods to determine the density of moist air. In: *IEEE Transactions on Instrumentation and Measurement* 52 (2003), Nr. 2, S. 504–507

[94] Pirc, J. ; Turk, G. ; Zura, M.: Highway travel time estimation using multiple data sources. In: *Intelligent Transport Systems (IET)* 10 (2016), Nr. 10, S. 649–657

[95] RABINER, L. ; JUANG, B.: An introduction to hidden Markov models. In: *IEEE ASSP Magazine* 3 (1986), Nr. 1, S. 4–16

[96] READ, I. ; COX, S.: Automatic Pitch Accent Prediction for Text-To-Speech Synthesis. In: *Eighth Annual Conference of the International Speech Communication Association*. Antwerp, Belgium : ISCA, 2007, S. 482–485

[97] REID, D.: *Ford and Europe's auto giants to build a long-distance electric charging network*. `https://www.cnbc.com/2017/11/03/ford-bmw-vw-daimler-building-electric-charging-network-twice-the-power-of-teslas.html`. Version: 2017. – accessed 22.04.2021

[98] RHODE, S.: *Robust and regularized algorithms for vehicle tractive force prediction and mass estimation*. Karlsruhe, Karlsruher Institut für Technologie, Dissertation, 2016

[99] RHODE, S. ; GAUTERIN, F.: Online Estimation of Vehicle Driving Resistance Parameters with Recursive Least Squares and Recursive Total Least Squares. In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. Gold Coast, QLD, Australia : IEEE, 2013, S. 269–276

[100] RHODE, S. ; HONG, S. ; HEDRICK, J. ; GAUTERIN, F.: Vehicle tractive force prediction with robust and windup-stable Kalman filters. In: *Control Engineering Practice* 46 (2016), S. 37–50

[101] RICE, J. ; VAN ZWET, E.: A simple and effective method for predicting travel times on freeways. In: *IEEE Transactions on Intelligent Transportation Systems* 5 (2004), Nr. 3, S. 200–207

[102] RODGERS, L. ; FREY, D. ; WILHELM, E.: Estimating an electric Vehicle's Distance to empty using both past and Future Route Information. In: *ASME International Design Engineering Technical Conference (IDETC)*. Portland, OR, USA : ASME, 2013, S. 1–9

[103] Rupnik, J. ; Davies, J. ; Bla, F. ; Duke, A. ; Clarke, S.: Travel Time Prediction on Highways. In: *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. Liverpool, UK : IEEE, 2015, S. 1435–1442

[104] Saffo, P.: *Six Rules for Effective Forecasting*. `https://hbr.org/2007/07/six-rules-for-effective-forecasting`. Version: Harvard Business Review 07/2007. – accessed 22.04.2021

[105] Salmasi, F.: Control Strategies for Hybrid Electric Vehicles: Evolution, Classification, Comparison, and Future Trends. In: *IEEE Transactions on Vehicular Technology* 56 (2007), Nr. 5, S. 2393–2404

[106] Sarrafan, K. ; Muttaqi, K. ; Sutanto, D. ; Town, G.: An Iintelligent Driver Alerting System for Real-time Range Indicator Embedded in EVs. In: *2016 IEEE Industry Applications Society Annual Meeting*. Portland, OR, USA : IEEE, 2016, S. 1–7

[107] Sarrafan, K. ; Sutanto, D. ; Muttaqi, K. ; Town, G.: Accurate range estimation for an electric vehicle including changing environmental conditions and traction system efficiency. In: *IET Electrical Systems in Transportation* 7 (2017), Nr. 2, S. 117–124

[108] Sautermeister, S. ; Falk, M. ; Bäker, B. ; Gauterin, F. ; Vaillant, M.: Influence of Measurement and Prediction Uncertainties on Range Estimation for Electric Vehicles. In: *IEEE Transactions on Intelligent Transportation Systems* 19 (2017), Nr. 8, S. 2615 – 2626

[109] Sautermeister, S. ; Ott, F. ; Vaillant, M. ; Gauterin, F.: Reducing Range Estimation Uncertainty with a Hybrid Powertrain Model and Online Parameter Estimation. In: *2017 IEEE 20th In-*

ternational Conference on Intelligent Transportation Systems (ITSC) (2017), S. 1–6

[110] SCHENNACH, Susanne M.: Convolution without independence. In: *Journal of Econometrics* 211 (2019), Nr. 1, S. 308 – 318

[111] SCHEUBNER, S. ; THORGEIRSSON, A. ; VAILLANT, M. ; GAUTERIN, F.: A stochastic Range Estimation Algorithm for Electric Vehicles using online Traffic Phase Classification. In: *IEEE Transactions on Vehicular Technology* 68 (2019), Nr. 7, S. 6414 – 6428

[112] SCHÖLKOPF, B. ; SMOLA, A.: *Learning with Kernels*. Cambridge, MA, USA : MIT Press Ltd, 2000. – ISBN 978–0262194754

[113] SCHRAMM, D. ; HILLER, M. ; BARDINI, R.: *Vehicle Dynamics: Modeling and Simulation*. Berlin, Germany : Springer, 2014. – ISBN 978–3540360445

[114] SEDDIG, K. ; JOCHEM, P. ; FICHTNER, W.: Integrating renewable energy sources by electric vehicle fleets under uncertainty. In: *Energy* 141 (2017), S. 2145–2153

[115] SHAWE-TAYLOR, J. ; CRISTIANINI, N.: *Kernel Methods for Pattern Analysis*. Cambridge, UK : Cambridge University Press, 2004. – ISBN 0–511–21237–2

[116] SHEIKHAN, M. ; BEJANI, M. ; GHARAVIAN, D.: Modular neural-SVM scheme for speech emotion recognition using ANOVA feature selection method. In: *Neural Computing and Applications* 23 (2013), Nr. 1, S. 215–227

[117] SILVA, M. ; GONCALVES, A. ; MARQUES, J. ; SOUSA, J. ; NUNES, U.: Modelling an Electric Vehicle Powertrain on Bench. In: *2014 IEEE Vehicle Power and Propulsion Conference (VPPC)*. Coimbra, Portugal : IEEE, 2014, S. 1–6

[118] SMOLA, A. ; SCHÖLKOPF, B.: A tutorial on support vector regression. In: *Statistics and Computing* 14 (2004), S. 199–222

[119] SOKEN, H. ; SAKAI, S.: A Multiple Model Adaptive Estimation Algorithm for Systems with Parameter Change. In: *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE).* Tsukuba, Japan : IEEE, 2016, S. 1718–1723

[120] SORET, A. ; GUEVARA, M. ; BALDASANO, J.: The potential impacts of electric vehicles on air quality in the urban areas of Barcelona and Madrid (Spain). In: *Atmospheric Environment* 99 (2014), S. 51–63

[121] STENLUND, B. ; GUSTAFSSON, F.: Avoiding windup in recursive parameter estimation. In: *Preprints of reglermöte 2002* (2002), S. 148–153

[122] STEPHENS, C. ; HUERTA, H. ; LINARES, A.: When is the Naive Bayes approximation not so naive? In: *Machine Learning* 107 (2018), Nr. 2, S. 397–441

[123] SU, Y. ; GAO, X. ; LI, X. ; TAO, D.: Multivariate Multilinear Regression. In: *IEEE Transactions on Systems, Man, and Cybernetics - PArt B: Cybernetics* 42 (2012), Nr. 6, S. 1560–1573

[124] TESLA INC.: *Tesla Model X.* `https://www.tesla.com/modelx`. – accessed 22.04.2021

[125] THORGEIRSSON, A. ; SCHEUBNER, S. ; F"UNFGELD, S. ; GAUTERIN, F.: An Investigation Into Key Influence Factors for the Everyday Usability of Electric Vehicles. In: *IEEE Open Journal of Vehicular Technology* 1 (2020), S. 348 – 361

[126] THORGEIRSSON, A. ; VAILLANT, M. ; SCHEUBNER, S. ; GAUTERIN, F.: Evaluating system architectures for driving range estimation and charge planning for electric vehicles. In: *Software: Practice and Experience* 51 (2021), S. 72–90

[127] TINGTING, D. ; JUN, L. ; FUQUAN, Z. ; YI, Y. ; QIQIAN, J.: Analysis on the influence of measurement error on state of charge estimation of LiFePO4 power Battery. In: *2011 International Conference on Materials for Renewable Energy and Environment*. Shanghai, China : IEEE, 2011, S. 644–649

[128] VAILLANT, M.: *Design Space Exploration zur multikriteriellen Optimierung elektrischer Sportwagenantriebsstränge*. Karlsruhe, Karlsruher Institut für Technologie, Dissertation, 08.10.2015

[129] VALADKHANI, A. ; HONG, Y. ; RAMEZANI, M.: Integration of Loop and Probe Data for Traffic State Estimation on Freeway and Signalized Arterial Links. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. Yokohama, Japan : IEEE, 2017, S. 1–6

[130] VANAJAKSHI, L. ; RILETT, L.: A Comparison Of The Performance Of Artificial Neural Networks And Support Vector Machines For The Prediction Of Traffic Speed. In: *IEEE Intelligent Vehicles Symposium*. Parma, Italy : IEEE, 2004, S. 194–199

[131] VAPNIK, V.: *Statistical Learning Theory*. Hoboken, NJ, USA : John Wiley & Sons, 1998. – ISBN 978–0–471–03003–4

[132] VAPNIK, V. ; GOLOWICH, S. ; SMOLA, A.: Support vector method for function approximation, regression estimation, and signal processing. In: *Advances in Neural Information Processing Systems* 9 (1997), S. 281–287

[133] VASEGHI, S.: *Advanced Digital Signal Processing and Noise Reduction*. Chichester, UK : John Wiley & Sons, 2008. – ISBN 978–0–470–75406–1

[134] WADOO, S. ; SYED, A. ; SOOD, V. ; ALI, S.: Prediction of Traffic Density from Wireless Cellular Data. In: *2016 IEEE 19th Interna-*

tional Conference on Intelligent Transportation Systems (ITSC). Rio de Janeiro, Brazil : IEEE, 2016, S. 575–580

[135] WAHL, Hans-Georg: *Optimale Regelung eines prädiktiven Energiemanagements von Hybridfahrzeugen*. Karlsruhe, Karlsruher Institut für Technologie, Dissertation, 10.06.2015

[136] WANG, H. ; LI, J. ; CHEN, Q. ; NI, D.: Speed-Density Relationship: from Deterministic to Stochastic. In: *The 88th Transportation Research Board (TRB) Annual Meeting*. Washington, DC, USA : National Academy of Sciences, 2009, S. 1–20

[137] WANG, Y. ; JIANG, J. ; MU, T.: Context-Aware and Energy-Driven Route Optimization for Fully Electric Vehicles via Crowdsourcing. In: *IEEE Transactions on Intelligent Vehicles* 14 (2013), Nr. 3, S. 1331–1345

[138] WIERMAN, M.: *An Introduction to the Mathematics of Uncertainty*. `https://www.creighton.edu/fileadmin/user/CCAS/programs/fuzzy_math/docs/MOU.pdf`. Version: 2010. – accessed 22.04.2021

[139] WILLMOTT, C. ; MATSUURA, K.: Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance. In: *Climate Research* 30 (2005), S. 79–82

[140] WU, C. ; HO, J. ; LEE, D.: Travel-time prediction with support vector regression. In: *IEEE Transactions on Intelligent Transportation Systems* 5 (2004), Nr. 4, S. 276–281

[141] XIONG, R. ; SUN, F. ; GONG, X. ; GAO, C.: A data-driven based adaptive state of charge estimator of lithium-ion polymer battery used in electric vehicles. In: *Applied Energy* 113 (2014), Nr. 1, S. 1421–1433

[142] YAUCH, P.: *Traffic Signal Control Equipment: State of the Art*. Washington, D.C., USA : Transportation Research, 1990. – ISBN 0–309–04917–2

[143] YI, Z. ; BAUER, P.: Adaptive Multi-Resolution Energy Consumption Prediction for Electric Vehicles. In: *IEEE Transactions on Vehicular Technology* 66 (2017), Nr. 11, S. 10515–10525

[144] YU, H. ; TSENG, F. ; MCGEE, R.: Driving Pattern Identification for EV Range Estimation. In: *2012 IEEE International Electric Vehicle Conference*. Greenville, SC, USA : IEEE, 2012, S. 1–7

[145] ZHANG, H.: The Optimality of Naive Bayes. In: *International Florida Artificial Intelligence Research Society Conference*. Miami Beach, FL, USA : AAAI Press, 2004, S. 562–567

[146] ZHANG, Y. ; WANG, W. ; KOBAYASHI, Y. ; SHIRAI, K.: Remaining Driving Range Estimation of Electric Vehicle. In: *2012 IEEE International Electric Vehicle Conference*. Greenville, SC, USA : IEEE, 2012, S. 1–7

[147] ZHAO, Y. ; CAI, Y. ; SONG, Q.: Energy control of plug-in hybrid electric vehicles using model predictive control with route preview. In: *IEEE/CAA Journal of Automatica Sinica* 1 (2018), S. 1–8

[148] ZHENG, B. ; HE, P. ; ZHAO, L. ; LI, H.: A Hybrid Machine Learning Model for Range Estimation of Electric Vehicles. In: *2016 IEEE Global Communications Conference (GLOBECOM)*. Washington, DC, USA : IEEE, 2016, S. 1–6

[149] ZOUBIR, A. ; KOIVUNEN, V. ; CHAKHCHOUKH, Y. ; MUMA, M.: Robust Estimation in Signal Processing: A Tutorial-Style Treatment of Fundamental Concepts. In: *IEEE Signal Processing Magazine* 29 (2012), Nr. 4, S. 61–80

Nomenclature

Acronyms

AE absolute error

ANOVA analysis of variance

AR auto-regressive

API application programming interface

ARIMA auto-regressive integrated moving average

BER balanced error rate

BEV battery-electric vehicle

BN Bayesian net

CAN controller area network

CLT central limit theorem

CRPS continuous ranked probability score

DBN dynamic Bayesian net

DNBC dynamic naive Bayesian classifier

ECF energy consumption forecast

EKF extended Kalman filter

ESP electronic stability program

ETA estimated time of arrival

FF free-flow

FLA fuzzy-logic algorithm

GPS global positioning system

HEV hybrid electric vehicle

HMM hidden Markov model

ICEV internal combustion engine vehicle

IIR infinite impulse response

J wide moving jam

KDE kernel density estimation

LS least-squares

MAE mean absolute error

MAP maximum a posteriori

MC Markov chain

MCS sequential Monte-Carlo simulation

MEDAE median absolute error

MEDPE median percentage error

MISO multiple-input-single-output

MLR multi-linear regression

MME multiple model estimation

MPE mean percentage error

MSE mean squared error

NBC naive Bayesian classifier

NN neural network

PE parameter estimation

PI prediction interval

RBA rule-based algorithm

RKF robust and windup-stable Kalman filter

RKFM RKF with MME

RLS recursive-least-squares

RMSE root mean squared error

RPA relative positive acceleration

SY synchronized-flow

SMC sequential Monte-Carlo simulation

SOC state of charge

SOH state of health

SVM support vector machine

TPC traffic phase classification

TRDB traffic and routing database

VP velocity prediction

WCRPS weighted continuous ranked probability score

WLS weighted-least-squares

Symbols

symbol	unit	description
$\hat{(\cdot)}$		estimation
$\bar{(\cdot)}$		average
$\tilde{(\cdot)}$		forecast
$\tilde{(\cdot)}^{\mathrm{d}}$		deterministic forecast
$\tilde{(\cdot)}^{\mathrm{s}}$		stochastic forecast
a	$\mathrm{m\,s^{-2}}$	longitudinal acceleration of the vehicle
a_{req}	$\mathrm{m\,s^{-2}}$	required acceleration to realize the velocity prediction
c_{w}	-	aerodynamic drag coefficient
d	km	distance
$d^{(x)}$	m	distance to the vehicle x
e		model error
f_{r}	-	rolling resistance coefficient
f_u	veh / h	traffic flow
g	$\mathrm{m\,s^{-2}}$	gravitational acceleration
h		hypothesis model for regression
$\mathbf{i}_{\mathrm{c}}$		measured CAN data
$\mathbf{i}_{\mathrm{t}}$		real-time TRDB data
k	-	navigation map segment
l	m	length
l_k	m	length of k
m	kg	vehicle mass
o	-	observation
$\mathbf{p}$		driving resistance parameter vector
p_{a}	%	attainability of a destination
q	%	battery state of charge
r		residuals
r_{dyn}	m	dynamic wheel radius
r_{xy}		correlation coefficient of variables x and y
s_x		uncertainty of variable x

t	s	time
$\mathrm{thw}^{(x)}$	s	time headway to vehicle x
t_k	s	time on k
u	$\mathrm{m\,s^{-1}}$	probe distribution
u_{current}	$\mathrm{m\,s^{-1}}$	traffic speed on the route at the beginning of the trip
u_{ideal}	$\mathrm{m\,s^{-1}}$	ideal traffic speed on the route
$\bar{u}_k$	$\mathrm{m\,s^{-1}}$	mean traffic speed on k
$\bar{u}_{k,f}$	$\mathrm{m\,s^{-1}}$	free-flow speed on k
v	$\mathrm{m\,s^{-1}}$	vehicle velocity
v_{lim}	$\mathrm{m\,s^{-1}}$	legal speed limit
$v_{\text{lim},k}$	$\mathrm{m\,s^{-1}}$	legal speed limit on k
$v_{\text{rel}}^{(x)}$	$\mathrm{m\,s^{-1}}$	relative speed to vehicle x
w	-	number of folds in cross-validation
x		model variable
y		model variable
z		model variable
A	$\mathrm{m^2}$	vehicle frontal area
A_i		node in a Bayesian net
$\mathbf{B}$		dynamic vehicle states vector
$E(\cdot)$		energy consumption model
E_{aux}	kWh	energy consumption of the auxiliaries
E_{b}	kWh	battery energy content
E_{c}	kWh	tractive energy consumption
$\tilde{E}_{c\|a=0}$	kWh	energy consumption forecast without acceleration force
$\tilde{E}_{c\|a=const}$	kWh	energy consumption forecast with constant acceleration force
F	-	F-test result quantity
F_{r}	N	driving resistance force
F_{acc}	N	acceleration force
FL	-	free lane detected
H	-	expected frequency

I	-	total number of classes
I_{bat}	A	battery current
I_{red}	$\text{kg}\,\text{m}^{-2}$	reduced moment of inertia
N	-	total number of observations o
O	-	observed frequency
$\text{o}^{(x)}$	-	vehicle x detected
P	kW	tractive power
P_{bat}	kW	battery power
Q		sampling distribution
R_i	Ω	internal battery resistance
RPA	$\text{m}\,\text{s}^{-3}$	relative positive acceleration
S		scoring rule
T_{ϕ_1,ϕ_2}	-	transition probability between traffic phases ϕ_1 and ϕ_2
U	V	battery voltage
X^2	-	result of the χ^2 test
Y		true distribution of a stochastic system
Z		forecast distribution
α	°	road gradient angle
ε		insensitivity of the SVM-loss function
ζ		kernel transformation function
η_{dyn}	-	dynamic powertrain efficiency
η_{stat}	-	static powertrain efficiency
Θ		loss function
θ		Bayesian net parameters
κ_k	m^{-1}	road curvature
κ_k	m^{-1}	road curvature on k
μ		mean value of a distribution
ξ		Monte-Carlo sample
π		input feature vector
Π_{A_i}		parent of node A_i in a Bayesian net
ρ	$\text{kg}\,\text{m}^{-3}$	air density
ρ_u	veh / km	traffic density
σ		standard deviation of a distribution

τ		particle number in sequential Monte-Carlo
Φ		cumulative distribution function of the standard normal distribution
ϕ	-	traffic phase