

THE POTENTIAL OF APPLYING MACHINE LEARNING FOR PREDICTING CUT-IN BEHAVIOUR OF SURROUNDING TRAFFIC FOR TRUCK-PLATOONING SAFETY

Irene Cara
Jan-Pieter Paardekooper
TNO Helmond
The Netherlands

Paper Number 17-0292

ABSTRACT

Truck platooning has great potential for reducing transport costs by lowering fuel consumption and increasing traffic efficiency. The short time headway between trucks in a platoon makes detecting the behaviour of other road participants essential for safety. Current safety controllers rely only on the traffic situation at the same instant, but accurate predictions of traffic behaviour are necessary to optimize the distance between the trucks and use the full potential of truck platooning in a safe way.

This study aims to show the potential of applying machine learning techniques to in-vehicle sensor data for predicting a cut-in manoeuvre by a passenger car. We have trained several algorithms, ranging from linear regression to Support Vector Regression and LSTM neural networks, on a dataset of naturalistic driving that contains 146 cut-ins. The results were compared to a benchmark of linear extrapolation under the assumption of a constant speed of the passenger car.

The results show that many machine learning algorithms are no viable alternative to the constant speed benchmark, with the exception of linear methods and Support Vector Regression. Further development of the Support Vector Regression algorithm in a direct-recursive hybrid forecast framework (dubbed dr-SVR) shows improvement of the error in the longitudinal distance and speed with more than 40% compared to the benchmark. Testing the trained algorithm on a truck platooning dataset shows an improvement of 15%.

The dr-SVR model has the potential to improve the safety of truck platooning by predicting the behaviour of passenger cars after a cut-in. More training data, especially including rare outliers and cut-ins representative for merges in a truck platoon, are needed to improve the accuracy and make the method suitable for application in safety controllers in the platooning trucks.

INTRODUCTION

In truck platooning two or more trucks are driving with short inter-vehicle distance to reduce fuel consumption and improve traffic efficiency [1]. These short distances can be accomplished in a safe way by using Vehicle-To-Vehicle (V2V) communication to inform about the intended behaviour of the lead truck to the other trucks in the platoon, that are operated automatically. At these levels of automation the drivers of the following trucks cannot be considered as fallback option in safety-critical situations, so operational safety (like collision avoidance) must be ensured by the automated driving functions.

The trucks in the platoon are equipped with on-board sensors to detect the surrounding traffic. Based on this information unsafe situations can be predicted and collisions can be avoided [3]. An important situation for operational safety is a cut-in of a passenger car in front of one of the trucks, which is the focus of this work. Early prediction of the behaviour of the car performing the cut-in will help to increase operational safety, because the controllers can use this additional information to anticipate the behaviour of the car, which is especially important in case the V2V communication fails.

Modern cars are equipped with sensors to detect the surroundings of the car for use of advanced driving assistance systems and automated driving systems. This means that more and more data of naturalistic driving become available. The aim of this work is to determine the potential of applying machine learning algorithms to these data to improve the prediction accuracy of the behaviour of cars during and after a cut-in. Such algorithms can be used in the controller of the following trucks to avoid hazardous situations caused by other road participants or V2V failures.

PROBLEM DEFINITION

The objective of this study is predicting the behaviour of a passenger car (hereafter the target) after a cut-in action. An example of such situation is shown in Figure 1, where a passenger car cuts in between two trucks in a platoon. Defining the moment in which the target enters the field of view of the sensors of the host vehicle (which is the

moment that the cut-in is detected) as t_1 , and assuming the cut-in is accomplished at $t_2 = t_1 + 4s$, the research question can be summarized as: *Is it possible to predict the longitudinal distance, lateral position, longitudinal speed and longitudinal acceleration in an interval of time that goes from t_1 to t_2 ?*

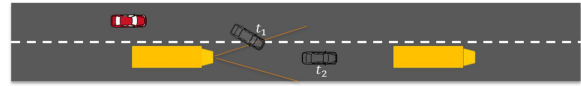


Figure 1. Cut-in action between two trucks in a platoon by a passenger car.

METHOD

The prediction algorithms were developed by training on vehicle-kinematics data of cut-ins that were extracted from naturalistic driving data. In this section we describe the datasets and the training procedure of the model.

Datasets

The training and validation of the prediction algorithm is done on a dataset of naturalistic driving of passenger cars that is part of the TNO Streetwise scenario database. This particular dataset describes a route of 48.5 km that was driven twice by 20 test persons, experienced drivers driving at least 5000 km/yr. The route took approximately 1 hour and 10 minutes to complete. The vehicle used was the TNO car lab Toyota Prius equipped with in-vehicle sensors as well as sensors looking at the environment like the radar and Mobileye system for lane detection. In addition to the in-vehicle signals, a video stream of the forward view and the GPS position of the vehicle was logged.

For testing the prediction algorithm, a truck platooning dataset was used. In this dataset the following truck in the platoon was equipped with a front-facing radar and camera. Thus all cut-ins recorded happened between the trucks in the platoon, as depicted in Figure 1.

Cut-in extraction

An internally developed cut-in detection algorithm was used for automatic extraction of cut-ins from the datasets. The intended application of truck platooning requires cut-ins on the highway, so we only considered cut-ins where the speed of the host vehicle was higher than 50 km/h. The start of the

cut-in is defined as the moment in which the target car crosses the lane marker. The behaviour of the target is followed for 4 seconds, which is the maximum prediction horizon of the algorithm. The relative position, absolute speed, and absolute acceleration of both the target car and the host vehicle are used as input for the prediction.

The training set contains 146 cut-ins which were used for training and validation of the learning models. In addition, 17 cut-ins from the truck platooning dataset were used for testing the trained model.

Reference model

The results of the prediction algorithm are compared to a baseline model that assumes that the speed of the target vehicle remains constant during and after the cut-in. For example, that means that the longitudinal distance at time t_i is given by

$$\Delta x(t_i) = \Delta x(t_0) + (t_i - t_0)(v_{\text{target}}(t_0) - v_{\text{host}}(t_0))$$

In Figure 2 we compare the predicted longitudinal distance of this model with the ground truth. We also report the root-mean-square error (RMSE) of the predictions, computed as

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

where y_i is the ground truth of the predicted value, and \hat{y}_i the prediction. The baseline model performs very well, showing that cars do not often change speed during a cut-in. After 1 second the average prediction error is 24 cm, while after 4 seconds the error has grown to 2.4 m. However, this model is not able to catch the outliers that are most interesting in terms of safety. For the prediction algorithm to be useful in practice it needs to perform better than this baseline and be able to predict outliers.

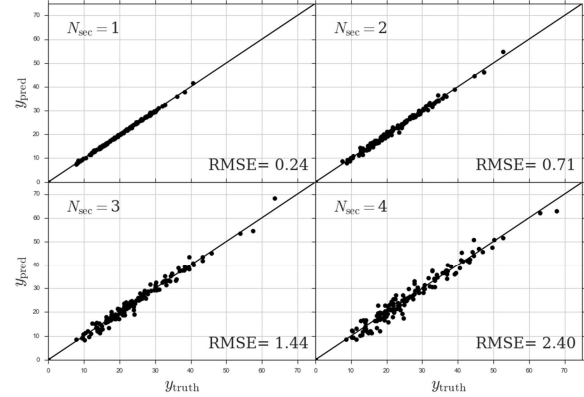


Figure 2. The prediction as function of the ground truth for the longitudinal distance as predicted by the linear model.

Forecasting strategy

Time series forecasting is a well-known problem in machine learning, used in many different fields (see [2] for a review). Often only a single time step ahead needs to be predicted. However, this study focusses on predicting the behaviour of the target car at multiple times in the future. Several strategies exist for this so-called multi-step forecasting, for example:

- **Direct forecast:** develop a separate model for every forecast time step.
- **Recursive forecast:** a one-step model that uses the output of the previous time step as input.
- **Multiple-output forecast:** a single model that is capable of predicting the entire forecast sequence at once.
- **Direct-recursive hybrid forecast:** a separate model for every forecast time step, that takes the output of the previous time step as input.

In the following we describe our experiments with different strategies to determine which method suits this problem best.

Machine learning models

Spot-checking In machine learning there is no one model that works best for every problem, the “no free lunch” theorem [4]. It is therefore important to perform spot-checking: quickly try out different machine learning algorithms without optimisation, to select the one that is best suited to the problem at hand for optimisation. In this study we have used several algorithms for direct forecasting to determine which one to use in direct-recursive forecasting.

The algorithms used are:

- **Linear regression:** Similar to ordinary least-squares fitting.
- **Ridge regression:** A linear model imposing a penalty on the size of the coefficients.
- **Lasso:** A linear model that estimates sparse coefficients.
- **K-nearest neighbours:** Prediction is based on the mean of the K most similar instances.
- **Decision trees:** Prediction is based on a binary tree model of the data.
- **Support vector regression:** Prediction is based on the idea that it is possible to build a hyperplane that can separate two different sets of objects.

The input for all these machine learning algorithms consists of the relative position, absolute speed, and absolute acceleration of both the target car and the host vehicle at the start of the cut-in.

LSTM neural network In addition to these direct forecasting methods, we have also applied a Long Short Term Memory (LSTM) neural network. These networks are designed to have a memory and are thus the form of deep learning that is best suited for time series. In contrast to the methods discussed above, LSTM neural networks can take multiple time steps as input. It is therefore a recursive forecasting method. As input we have used the time vector in the first second of the cut-in for the same set of parameters described above. Because the number of cut-ins is limited, we used a small network of 4 neurons, bigger networks would overfit and produce less accurate results.

Training All algorithms were trained on the passenger car dataset using leave-one-out cross validation. In this procedure, the models are trained on $(N - 1)$ samples from a training set of N samples, and the prediction error of the model on the remaining sample is computed. This procedure is repeated N times. As error measure for the prediction error we used the RMSE (Equation 1).

RESULTS

Direct forecast

In Figure 3 we show the results of the direct forecasting methods. For clarity the focus is on the prediction of the longitudinal distance only.

With the exception of the Lasso, the linear methods (linear regression, ridge regression) perform very well, with an average error less than 2 meter for a prediction horizon of 4 seconds. This is not surprising since the naive constant speed model, that is also a linear model for the distance, shows similar results. Almost all the non-linear methods do not perform as well and have much higher errors than the constant speed model. The positive exception is the SVR model, that shows performance equal to linear regression. Because (in contrast to linear regression) SVR is capable of catching non-linear behaviour, we chose this algorithm for further development in the direct-recursive framework.

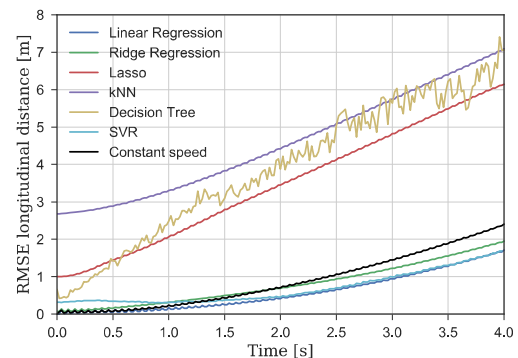


Figure 3. The RMSE in the longitudinal distance as function of time for prediction using linear regression, ridge regression, lasso, k-nearest neighbours, decision tree, support vector regression and the constant speed model.

LSTM neural network

In Figure 4 we show the prediction error for the LSTM network. The network has as input the time series of the input between 0 and 1 second, hence the forecasting starts at 1 second. Despite this additional information, the LSTM network performs worse than the three best direct forecasting methods. Most likely this has to do with the limited number of cut-ins in the training set. Neural networks require more data than other methods for optimal performance. For this reason we decided to focus on the direct-recursive method instead.

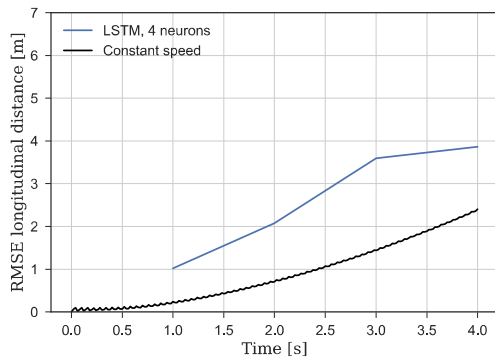


Figure 4. The RMSE in the longitudinal distance as function of time for prediction using an LSTM neural network consisting of 4 neurons.

Direct-recursive Support Vector Regression

In the direct forecasting results the Support Vector Regression (SVR) algorithm shows the most promising performance. Here we show the results of using SVR as a direct-recursive hybrid forecasting algorithm: dr-SVR. For every prediction time step we train a separate SVR that takes as input the output of the previous time step. We trained four separate dr-SVR models for prediction of the longitudinal distance, the lateral distance, the longitudinal speed and the longitudinal acceleration. The results of the vanilla dr-SVR were further improved by optimising the penalty term for misclassifications, the kernel scale and the error distance.

Prediction error Figure 5 shows the prediction error in the longitudinal distance to the host vehicle, the lateral distance to the lane marker, the longitudinal speed and the longitudinal acceleration of the target car as function of time for the passenger car data, for both the optimised and non-optimised dr-SVR. For reference we also show the baseline model and the direct forecast of the SVR.

The optimised dr-SVR outperforms all the other models, reducing the error with 48%, 26%, 44% and 19% for respectively the longitudinal distance, lateral distance, speed and acceleration, as compared to the baseline model at a prediction horizon of 2 seconds. The dr-SVR results show an improvement of around 30% compared to direct forecasting with SVR.

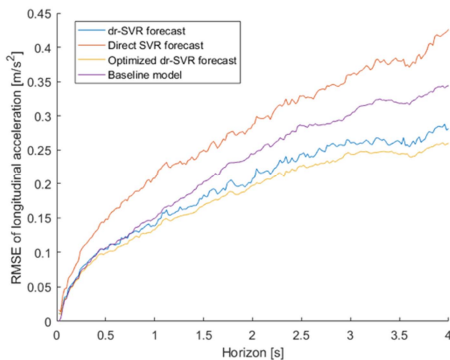
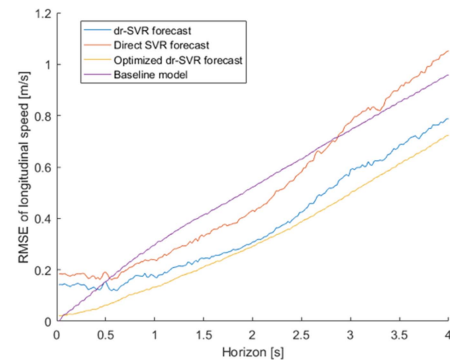
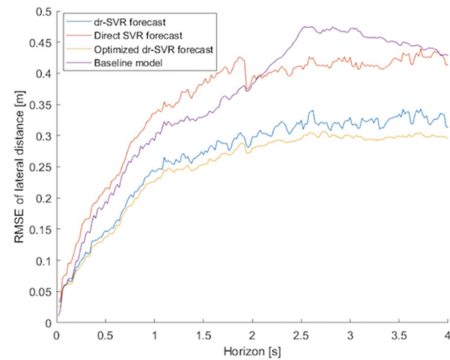
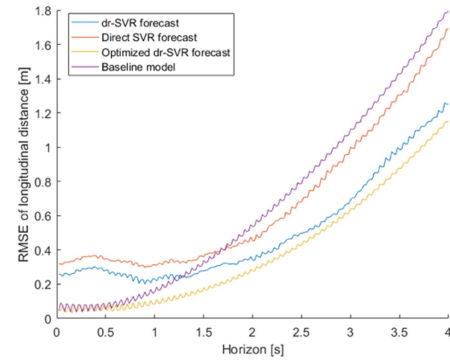


Figure 5 The RMSE of the prediction of the longitudinal distance, the lateral distance, the longitudinal speed and the longitudinal acceleration with dr-SVR (both optimised and non-optimised) as compared to direct forecasting with SVR and the baseline model, for the passenger car dataset.

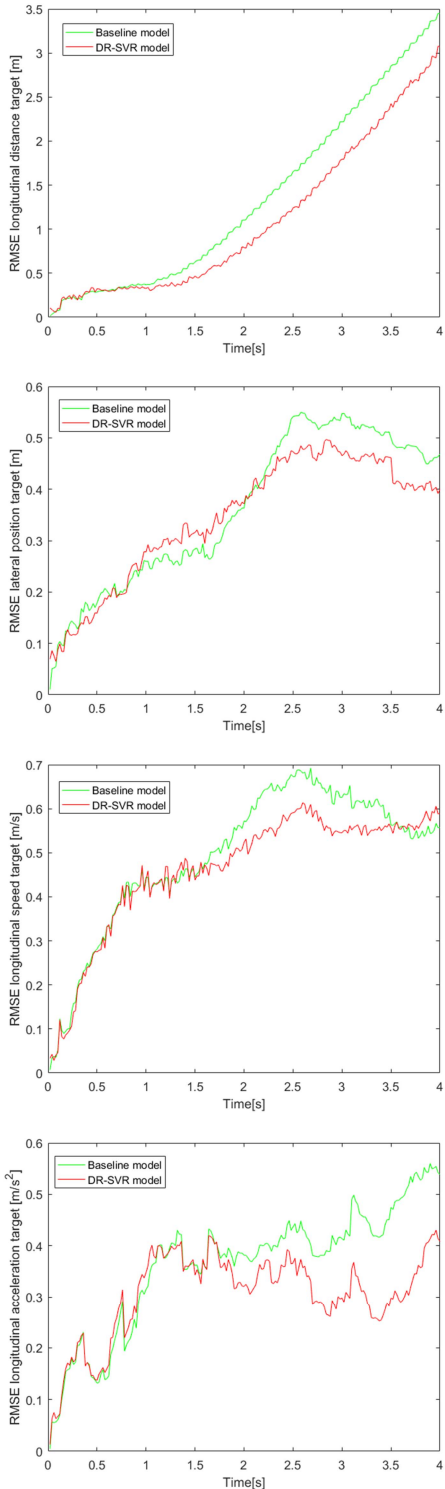


Figure 6. The RMSE of the prediction of the longitudinal distance, the lateral distance, the longitudinal speed and the longitudinal acceleration with dr-SVR compared to the baseline model, for the truck platooning dataset.

To validate the dr-SVR model for the intended application of truck platooning, we applied the trained model to 17 cut-ins from the truck platooning dataset, without re-training the algorithm. The truck platooning dataset does not contain information on the lane markers, therefore the model was changed in order to predict the relative position to the truck and not the lane marker. Figure 6 shows the RMSE of the forecast of the longitudinal and lateral distance to the host truck, the longitudinal speed and the longitudinal acceleration of the target car. The improvement of the dr-SVR model compared to the baseline is less evident, but still present, especially in the forecasting of the longitudinal distance and speed: the dr-SVR shows an improvement of the prediction of both the longitudinal distance and longitudinal speed of up to 15%.

Prediction examples

Figure 7 shows an example of a prediction for a typical cut-in. Although the speed during the cut-in is not far from a linear extrapolation, the dr-SVR is able to reduce the error in the prediction by predicting the non-linear behaviour correctly, especially from 2.5 to 4 seconds. After 4 seconds, the error of the dr-SVR prediction in the longitudinal distance is 1.5 m, in the lateral distance 0.25 m, in the longitudinal speed 0.05 m/s and in the acceleration 0.15 m/s².

In Figure 8 we show an example of an outlier where the prediction of the dr-SVR has large errors. This figure shows an atypical cut-in where the target car accelerates after 2.5 seconds. This behaviour is not picked up by the dr-SVR, resulting in large prediction errors after 4 seconds.

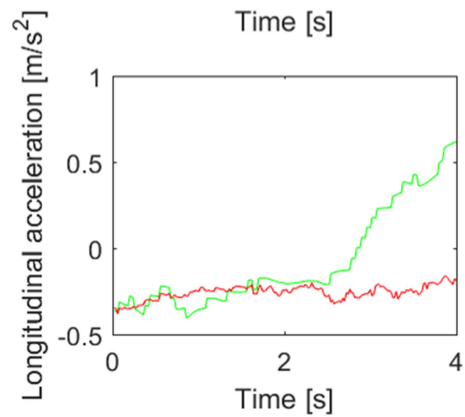
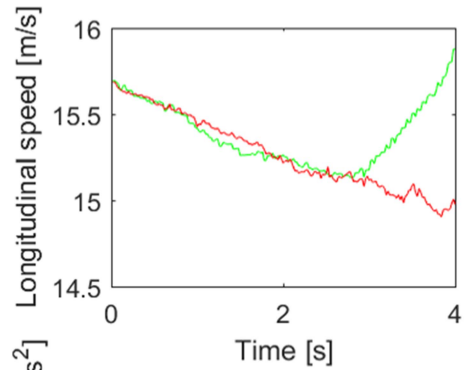
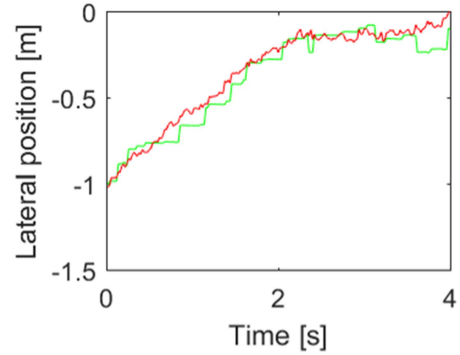
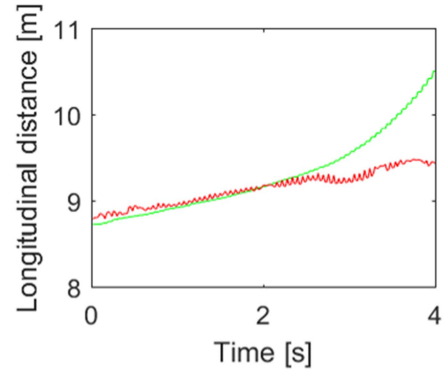
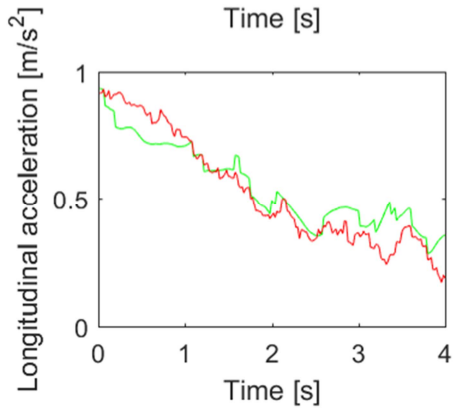
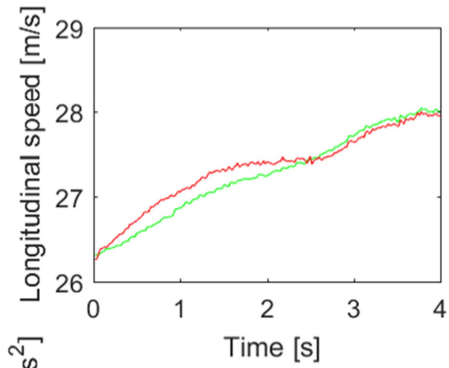
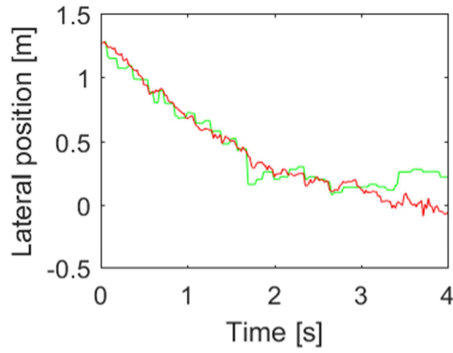
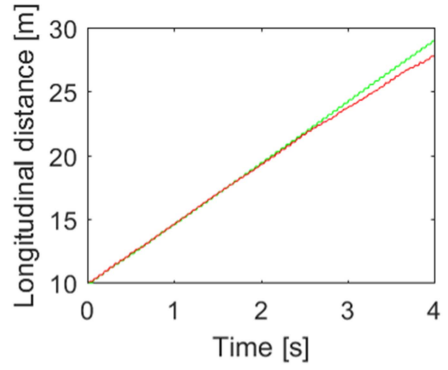


Figure 7. Example of the prediction of the longitudinal distance, lateral distance, longitudinal speed and longitudinal acceleration for a typical cut-in in the passenger car dataset. The ground truth is shown in green, while the prediction is shown in red.

Figure 8. Example of the prediction of the longitudinal distance, lateral distance, longitudinal speed and longitudinal acceleration for an outlier in the passenger car dataset. The ground truth is shown in green, while the prediction is shown in red.

DISCUSSION

The results of this study show that in a typical cut-in the behaviour of the target car is well-approximated by linear extrapolation of the longitudinal speed. The dr-SRV is able to improve on this by predicting small non-linearities. This shows that the dr-SVR is a promising algorithm for behaviour prediction after a cut-in. However, the improvement of the dr-SVR in the prediction of outliers is small. The reason is the few training examples of a-typical cut-ins. In order to catch this behaviour, the algorithm needs to be trained on more of these cut-ins. Given the ability of the dr-SVR to predict small non-linearities in the behaviour, we expect a substantial improvement in prediction accuracy of outliers with more training examples.

The trained dr-SVR generalises quite well to truck platooning data, although the prediction accuracy is less than for the passenger car data. This is due to the difference in cut-ins that occur when a passenger car merges with a truck platoon: the space for the manoeuvre is smaller, and usually the target car performs a cut-through manoeuvre to take an exit road on the highway. To include this kind of behaviour, the model should be trained with this kind of cut-ins as well.

CONCLUSIONS

The safety of truck platooning can be improved by predicting the behaviour of the surrounding traffic. In this study the potential of machine learning for the development of a cut-in prediction algorithm was determined. Our conclusions can be summarised as follows:

- Linear extrapolation assuming a constant longitudinal speed predicts the behaviour in typical cut-ins well. Many direct forecasting methods and LSTM neural networks are not able to improve on this baseline.
- The dr-SVR is able to accurately predict small non-linearities in the target car behaviour, thus improving the prediction accuracy of the linear benchmark. We expect that with more training data, the accuracy of predicting outliers will improve substantially.
- The dr-SVR generalises well to truck platooning data, although the performance

is expected to improve when more training data is available.

For application of the cut-in prediction algorithm in the safety controllers of a truck platoon, the accuracy needs to be improved. Collecting more cut-in data, especially outliers, to better train the dr-SVR model is one way of reducing the prediction error. Further improvement of the prediction is expected by taking the kinematics of the target into account. Instead of predicting the longitudinal distance, speed, and acceleration with separate dr-SVR models, the relationship between distance, speed, and acceleration can be taken into account to further improve the accuracy of the prediction. As a separate application we aim to extend the algorithm to include early prediction of cut-in intention: predicting a few seconds beforehand, when the target car is still in the other lane, that a cut-in is going to happen.

ACKNOWLEDGEMENT

The authors would like to thank Riske Meijer for useful comments on the initial draft of this manuscript and the teams involved in EcoTwin and the Early Research Project Human Enhancement for providing the measurement data.

REFERENCES

- [1] Janssen, R. and Zwijnenberg, H. and Blankers, I. and de Kruijff, J. 2015. "Truck platooning driving the future of transportation."
- [2] Shumway, R. H. and Stoffer, D. S. 2010. "Time Series Analysis and Its Applications." Springer.
- [3] Van Nunen, E. and Elfring, J. and Uittenbogaard, J. and Ploeg, J. 2016. "Applying V2V for operational safety within truck platooning." In Proceedings of the 23rd ITS World Congress (Melbourne, Australia, Oct 10-14).
- [4] Wolpert, David (1996), "The Lack of A Priori Distinctions between Learning Algorithms", Neural Computation, pp. 1341-1390.