2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 11-14, 2020. Toronto, Canada

A Hybrid Framework Combining Vehicle System Knowledge with Machine Learning Methods for Improved Highway Trajectory Prediction

Manuel Muñoz Sánchez[†§], Emilia Silvas[*§], Denis Pogosov[§], Decebal Constantin Mocanu[†‡]
[†]Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands
[*]Department of Mechanical Engineering, Eindhoven University of Technology, The Netherlands
[§]Department of Integrated Vehicle Safety, TNO, The Netherlands
[‡]Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, The Netherlands
Email: {manuel.munozsanchez, emilia.silvas, denis.pogosov}@tno.nl, d.c.mocanu@utwente.nl

***Abstract*—Vehicle-to-vehicle communication is a solution to improve the quality of on-road traveling in terms of throughput, safety, efficiency and comfort. However, road users that do not communicate their planned activities can create dangerous situations, so prediction models are needed to foresee and anticipate their motions in the drivable space. Various prediction methods exist, either physics-based, data-based or hybrids, but they all make conservative assumptions about others' intentions, or they are developed using unrealistic data, and it is unclear how they perform for trajectory prediction. In this work, we introduce and demonstrate an optimal hybrid framework that overcomes these limitations, by combining the predictions of several physics-based and data-based models. Using on-road measured data we show that this novel framework outperforms the individual models in both longitudinal and lateral position predictions. We also discuss the required prediction boundaries from a safety perspective and estimate the accuracies of the models in relation to automated vehicle functions. The results achieved by this method will enable increased safety, comfort and even more proactive reactions of the automated vehicles.**

***Index Terms*—Autonomous driving, highway trajectory prediction, hybrid, system knowledge.**

I. Introduction

Cooperative Adaptive Cruise Control (CACC) is an extension of Adaptive Cruise Control that uses vehicle-to-vehicle (V2V) communication to share intentions and form a platoon of vehicles that can drive safely and save driver efforts [1]. Having V2V communication allows one to be immediately aware of the preceding vehicle's intentions, and has benefits such as avoiding ghost traffic jams, reducing CO_2 emissions, and enabling higher traffic throughput on the same infrastructure [2].

When several V2V-enabled vehicles are platooning, the inter-vehicle time gap is smaller than in typical ACC systems (i.e. 1.5 sec), and having other road users interfere with this platoon can lead to dangerous situations. To anticipate dangerous maneuvers, e.g. cut-ins (nearby lane changes), various prediction algorithms exist, either *physics-based* (PB), describing the vehicle's motion with kinematic equations [3], *data-based* (DB), extracting vehicle's behavior from past observations [4], or a combination of both [5], referred to as *hybrids*. PB methods are based on simple assumptions, such as constant velocity or acceleration, and perform best in short prediction horizons (e.g. < 1 second), while DB methods can capture non-linear or more complex on-road behaviors and perform better in longer term prediction. Yet, no method exists that offers qualitatively good results for short and long term predictions. Moreover, it is unclear how well these methods can predict trajectories when naturalistic driving data is considered, and there is no established quantification of which longitudinal and lateral prediction accuracies should be achieved for different prediction horizons.

State of the art prediction methods either first classify the type of maneuver (e.g. lane keeping, lane changing) and select the most appropriate regression model, or use some deep learning architecture that allows for direct regression without explicitly making this distinction. However, these methods usually have one or more of the following limitations: (i) assume correct classification of target vehicle's maneuver; (ii) use idealistic data for the development and evaluation of their methods, which is not realistic; and (iii) use partially erroneous or incomplete data. For instance, the Next Generation Simulation (NGSIM) dataset has become the basis of many studies [4], [6]–[10], despite it not being representative of all driving behavior and containing unrealistic relationships that are beyond repair applying post-processing techniques [11].

To overcome the aforementioned limitations, we propose a hybrid framework for vehicle trajectory prediction that effectively combines PB and DB models to produce more accurate predictions, making no assumptions on other road user's intentions, and using real on-road vehicle's data. The proposed hybrid framework can be directly incorporated in the architecture of a cooperative driving application, as shown in Fig. 1, to better anticipate the trajectory of non-communicative vehicles and react accordingly to dangerous maneuvers.

Key Contributions.: With our work, we present two main contributions: (1) a thorough comparison of PB and DB models for trajectory prediction on the highway, allowing for a better understanding of the capabilities and limitations of these methods when using naturalistic driving data; and

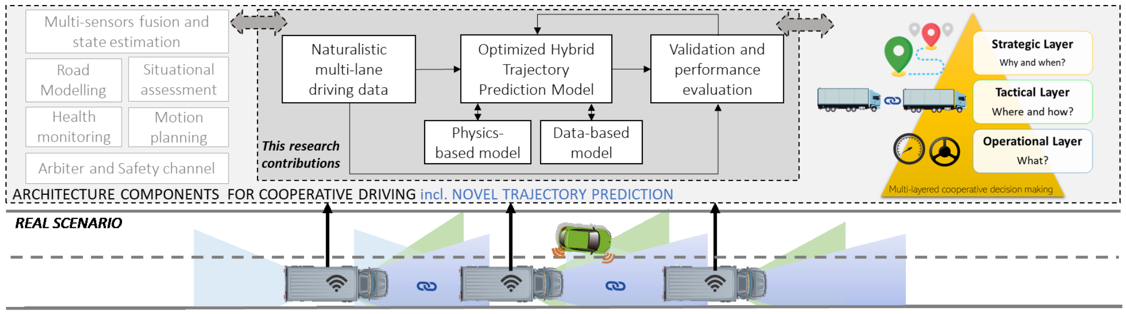


Fig. 1: Overview of the components of our cooperative driving application for truck platooning. The proposed hybrid framework can be directly integrated in the system architecture for improved trajectory prediction and anticipation of non-communicative vehicle maneuvers.

(2) a hybrid framework that presents improved trajectory prediction accuracy without prior knowledge of the type of maneuvers performed by other road users, which can be directly incorporated in a cooperative driving application for anticipation of dangerous maneuvers (see Fig. 1).

A. Problem Formulation

To develop a hybrid trajectory prediction framework that combines several models and enables more accurate predictions, first the motion of a vehicle needs to be described. For this problem, all data is measured from on-board sensors of the ego vehicle. Let the state of a vehicle, s, at any given time, t, be characterized by a collection of variables,

$$s(t) = (x, y, \theta, \omega, v, a), \tag{1}$$

where x and y are the longitudinal and lateral positions of the tracked vehicle; θ and ω are the heading angle and its rate of change; and v and a are the velocity and acceleration, all relative to the ego vehicle, as illustrated in Fig. 2. Furthermore, let $s(t_1, t_2)$ denote the states of the vehicle from time t_1 until time t_2, and $\hat{s}_m(t_1, t_2)$ the predictions of model m between t_1 to t_2. Similarly, let $s_f(t_1, t_2)$ denote the real, measured value of variable f, for any $f \in \{x, y, \theta, \omega, v, a\}$, and $\hat{s}_{f,m}(t_1, t_2)$ the predictions of model m for variable f.

Given a set of state observations T_p seconds in the past, $s(t_{now-T_p}, t_{now})$, predictions for the next T_{fut} seconds can be made, $\hat{s}_{m_i}(t_{now}, t_{now+T_{fut}})$ by a collection of prediction models, some of which may be PB, and some DB.

Finally, given a collection of prediction models, M, and an error metric, $\epsilon(s, \hat{s}_{m_i})$, used to quantify the deviation of predictions of model $m_i \in M$ from the ground truth, we wish to define a hybrid model H consisting of a combination of models from M, such that $\epsilon(s_{(x,y)}, \hat{s}_{(x,y)})$ is minimized.

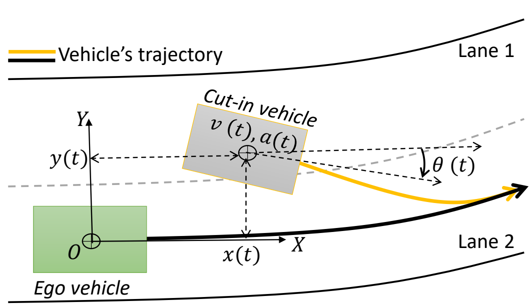


Fig. 2: Example of a dangerous lane change maneuver, where a vehicle on another lane performs a cut-in in front of the ego vehicle.

II. Background on Trajectory Prediction

To obtain a better understanding of the possibilities of developing hybrid prediction models, an overview of the main categories of forecasting models that can be used for trajectory prediction is given in this section. Although not all of them are currently being applied in the context of autonomous driving, we distinguish three main categories: physics-based, data-based, and hybrid models.

A. Physics-based models

In PB models, the behavior of the system is modeled using known kinematic equations, in which their parameters usually have physical meaning (e.g. the equation describing the traveled x for a certain t). Their main advantage is that they are interpretable. In the context of vehicle motion prediction, various models exist, each with different assumptions. Several of the most commonly used PB models are compared in [3], such as Constant Velocity (CV), and Constant Turn Rate and Acceleration (CTRA), with the latter being the preferred model considering both accuracy and computational cost. In the CV model, the vehicle state s at time t is represented by

$$s(t) = (x, y, \theta, v), \tag{2}$$

and the transition equation to compute the state of the vehicle Δt seconds in the future is given by

$$s(t + \Delta t) = s(t) + (v \cdot t \cos(\theta), v \cdot t \sin(\theta), 0, 0). \tag{3}$$

In the CTRA model, the state space is represented as

$$s(t) = (x, y, \theta, v, a, \omega), \tag{4}$$

and the transition equation is given by

$$s(t + \Delta t) = s(t) + (\Delta x, \Delta y, \omega \cdot t, a \cdot t, 0, 0), \tag{5}$$

where Δx and Δy are given by

$$\Delta x = \frac{v + at}{\omega} \sin(\theta + \omega t) + \frac{a}{\omega^2} \cos(\theta + \omega t) - \frac{v}{\omega} \sin(\theta) - \frac{a}{\omega^2} \cos(\theta), \tag{6}$$

$$\Delta y = -\frac{v + at}{\omega} \cos(\theta + \omega t) + \frac{a}{\omega^2} \sin(\theta + \omega t) + \frac{v}{\omega} \cos(\theta) - \frac{a}{\omega^2} \sin(\theta). \tag{7}$$

B. Data-based models

With DB models, the behavior of the system is considered unknown a priori, and an approximation is inferred from past observations. These models can approximate complex behavior, and therefore tend to be more accurate than PB models for longer prediction horizons, $N \in (1,3)$ seconds. However, the quality of the models is heavily affected by the quality and quantity of available observations (e.g. completeness, or noise). Moreover, DB models could yield unrealistic (e.g. physically impossible) predictions of the system state. Within DB models, we distinguish two main categories: *classical statistical* methods and *machine learning* methods.

a) Classical statistical methods: Examples of these methods include Exponential Smoothing [12] or ARIMA [13]. Typically, these methods are more interpretable than machine learning approaches, especially the ones based on large deep Artificial Neural Networks (ANN). However, they require statistical knowledge to apply them appropriately. Some approaches exist to automate forecasting with ARIMA models and exponential smoothing methods [14], [15], for single-step, univariate forecasting, but it is unclear they perform for multi-step, multivariate forecasting. This makes them unappealing for our problem.

b) Machine Learning methods: Two main types exist depending on the nature of the problem: *classification* or *regression*. In the context of vehicle motion prediction, *classification* is done computing the probability of a maneuver (e.g. probability that a vehicle will turn left). On the other hand, *regression* is done to predict the value of system state variables for some time in the future (e.g. vehicle trajectories [4]).

C. Hybrid models

Hybrid models combine the predictions of multiple models, benefiting from their strengths, but developing an optimal combination is usually challenging. Hybrids are analogous to ensemble methods [16], although ensemble methods is a term used when referring to the combination of multiple DB models. Thus, in this work the term *hybrids* is used to emphasize that the model is a combination of PB and DB models. Hybrid methods found in literature can be categorized as follows.

a) Hard Constraining: Multiple models predict variables of interest and the predictions are combined. This category can be divided in two sub-categories depending on the variables predicted by each model: (i) *direct*, when all the individual models predict all the variables of interest [5]. The individual predictions can then be aggregated, using for instance Kalman filtering [17]; and (ii) *piece-wise*, when different models predict different variables, e.g. some are predicted using a PB model and others using a DB model [18].

b) Soft Constraining: A main model's predictions are influenced by secondary models. For instance, PB neural networks: a physics-based term is added to the loss function of a neural network to penalize unrealistic predictions [19], [20].

c) Residual Modeling: The predictions of a main model are complemented by secondary models, which are used to model, predict, and correct the main model's errors. Several combinations are possible depending on the assumptions made about the system and the residuals, the type of models used, and how predictions are combined. Table I presents an overview of residual modeling hybrids found in literature.

TABLE I: Residual modeling hybrids found in literature, with different assumptions, individual models, and prediction combinations.

Model Main	Secondary	**Assumptions** System	Residuals	**Comb-ination**	**Reference**
PB	DB	L/NL	NL	L	[21]
DB (DBN)	DB (ARIMA)	NL	L	L	[22]
DB (ARIMA)	DB (ANN)	L	NL	L	[23]–[25]
DB (ARIMA)	DB (SVR)	L	NL	NL	[26]

L - Linear | NL - Non-linear | DBN - Deep Belief Network | SVR - Support Vector Regression

III. Constructing the Hybrid Framework

To predict the trajectories of non-communicative vehicles and integrate them in platooning, the proposed framework, its main components and its integration in the higher picture (application) are depicted in Fig. 1. As shown in [5], the hybridization of PB and DB approaches can benefit for short prediction horizons, $N \in (0,1)$ seconds, from the accuracy of the known vehicle dynamics, and for longer prediction horizons, $N \in [1,3]$ seconds, from the complexity that DB models can capture. Following the same reasoning, to make our hybrid, we first choose suitable PB and DB components.

a) Dataset: To benchmark the different methods and develop an improved hybrid framework, naturalistic driving data collected using our own ego vehicle equipped with several sensors is used. This data, measured $s(t)$ from Equation (1), contains 5-second trajectories of target vehicles, out of which we consider $T_p = 2$ and $T_{fut} = 3$ for the observations and predictions, respectively. Furthermore, maneuvers performed in these trajectories may be of two types, but a majority are lane keeping. To investigate the impact of optimizing our approach with different types of maneuvers, four data splits are created: *keeps*, containing trajectories where the tracked vehicle stays on the same lane; *changes*, containing trajectories where the tracked vehicle changes lanes; *imbalanced*, containing a majority of lane keeping trajectories; and *balanced*, containing the same number of lane keeping and lane changing trajectories. The four data splits are summarized in Table II. These splits are divided into training (90%) and testing (10%).

TABLE II: Different splits created from the available trajectories.

data split	# trajectories	% keeps	% changes
keeps	28564	100	0
changes	2546	0	100
imbalanced	31110	91.82	8.18
balanced	5092	50	50

b) Safety- and comfort-related performance metrics: In platooning applications, as depicted in Fig. 1, the automated vehicles drive close to each other, i.e. smaller time gaps, and any other vehicle interfering with the string of vehicles can pose a threat to the safety of the platoon. Detecting a cut-in late can lead to harsh braking and driver discomfort, or even to unsafe situations [1]. Although the need of having trajectory prediction algorithms has been acknowledged [4], so far no definitions exist of what is a good prediction error for an automated vehicle to react safely and comfortably to other vehicles. For example, intuitively it is more relevant to have a good prediction 1 second ahead than 3 seconds ahead. To clearly define these horizon-changing metrics for longitudinal/lateral accuracy, the motion planning, control algorithms, and the vehicle need to be in the loop. In this work, we do not close this loop, we focus on highway driving data where relatively high velocities are driven. For this purpose, we define the acceptable error boundaries as 2 meters in longitudinal distance and 0.45 meters (1/8 of a typical lane width) in lateral distance. Furthermore, to assess the performance of the models we evaluate, we look at a large range of N values, $0 < N \leq 3$.

A. Hybrid Components

1) Physics-based component: To understand which PB component is suitable to predict vehicle trajectories, we compare two commonly used models [3]: CV and CTRA, as introduced by (2)-(7). When using the CTRA model, the turn rate, ω, can be zero or very small. When ω is close to zero ($\omega < 10^{-4}$), (6) and (7) are replaced by

$$\Delta x = \cos(\theta)(v \cdot t + \frac{1}{2}a \cdot t^2), \tag{8}$$

$$\Delta y = \sin(\theta)(v \cdot t + \frac{1}{2}a \cdot t^2), \tag{9}$$

to avoid zero division issues. When using realistic driving data, as in our case, the noise in the data can make the predictions very inaccurate. This noise is especially problematic for variables $f \in \{\theta, \omega, v, a\}$. To compensate for this, the initial state of the target vehicle for each variable f, s_{0_f}, is smoothed using a weighted average of the most recent observations, with more recent observations having a higher weight, given by

$$s_{0_f} = \begin{cases} s_f(t_{now}) & f \in \{x, y\} \\ \frac{1}{5.5}\sum_{i=0}^{i=9}(1 - \frac{i}{10})s_f(t_{now-i \cdot r}) & f \in \{\theta, \omega, v, a\} \end{cases} \tag{10}$$

where r denotes the inverse of the data sampling rate, $r = 12.5\text{Hz}^{-1} = 0.08$ seconds. Longitudinal and lateral position are excluded from the smoothing process to use the latest sensed target vehicle position.

Both CV and CTRA models are evaluated on the four data splits. Independently of the type of maneuver at hand, on average CV outperforms CTRA. This behavior is not out of the ordinary for lane keeping trajectories, where the main assumption is that the vehicle drives straight. However, for lane changing trajectories one would expect CTRA to perform better, which is not the case here due to the unreliability of θ and ω in the data. Fig. 3 shows their performance on the balanced test set. For longitudinal position, after $N = 1$, CTRA's average longitudinal prediction errors, $\epsilon_{x,\mu}$, rapidly increase compared to CV, which is to be expected since CTRA assumes the target vehicle keeps turning. Furthermore, for the lateral position average error, $\epsilon_{y,\mu}$, the difference is small yet significant, with higher variance after 1.5 seconds. Overall, on realistic data, CV shows more accurate results.

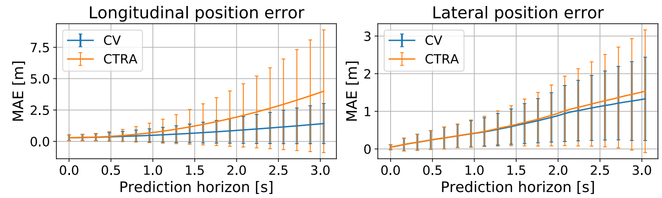


Fig. 3: Mean Absolute Error (MAE) of CV and CTRA predictions for trajectories of the balanced test set. The vertical lines indicate the standard deviation of the prediction errors.

2) Data-based component: To choose a DB model suitable for vehicle motion prediction, the nature and complexity of the maneuvers need to be considered. For this task, the authors of [4] and [27] show that machine learning methods, such as Long Short-Term Memory (LSTM) ANNs show potential. This model is also chosen here since LSTMs are well known for sequence to sequence mapping [28], hence also suitable for our trajectory prediction problem, i.e. mapping a sequence of (past) vehicle states to another sequence of (future) vehicle states. In this work, an LSTM as shown in Fig. 4 is implemented and compared to a Multilayer Perceptron (MLP).

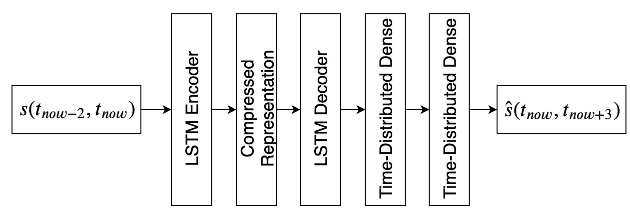


Fig. 4: Overview of the LSTM encoder-decoder architecture.

Althought MLPs are not particularly well suited for temporal data, we wish to verify that our LSTM model outperforms a standard MLP, which despite being a basic deep learning architecture, it is still widely used in practice (61% of the workload in Google TPUs [29]). Since MLPs do not support the extra temporal dimension of the data by default, our model is developed to accept several inputs (entire time window of state observations for each feature), and produces multiple outputs (entire window of state predictions for each predicted feature), as shown in Fig. 5.

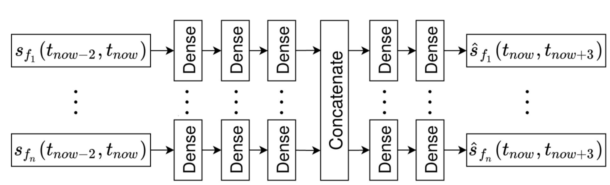


Fig. 5: Overview of the MLP architecture.

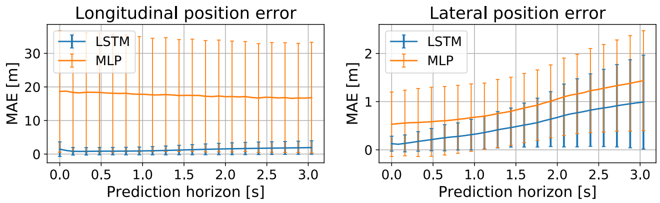


Fig. 6: MAE of MLP and LSTM predictions for trajectories of the balanced test set. Both models were trained on the balanced set.

A thorough comparison of the MLP and LSTM's accuracy was performed and it is clear that the LSTM is better suited for this prediction task. As shown in Fig. 6, the non-recurrent architecture of the MLP is completely unable to predict the target vehicle's longitudinal position accurately with $\epsilon_{x,\mu} = 20$ meters, and great variance. On the other hand, the difference in $\epsilon_{y,\mu}$ with respect to the LSTM is not as large, but it is unacceptable since a typical lane width is 3.6 meters. Following this analysis, the LSTM is considered more suitable than the MLP as the DB component of our hybrid framework.

Now that we have chosen a suitable DB component, we perform an extensive analysis of the impact of training it with maneuver-specific (*keeps* or *changes* data splits) vs. mixed (*imbalanced* or *balanced* splits) trajectories. To predict lane keeps, it is beneficial to train the model with some lane changes, especially for longitudinal position prediction when $N \geq 1$ second. By introducing noise as lane changes, the model's generalization capabilities improve, preventing overfitting. However, if too many lane changes are introduced, the model's performance decreases significantly (Fig. 7). Similarly, to predict lane changes it is beneficial to include some lane keeping trajectories. For lateral position prediction, this only holds for $N < 1$ second, while for longitudinal position predictions it holds even for $1 \leq N \leq 3$ (Fig. 8).

The performance of the LSTM trained on *changes* when predicting lane keeps is very poor compared to the LSTMs trained on either *balanced* or *imbalanced* (Fig. 9). This is expected since the model has not seen any lane keeping

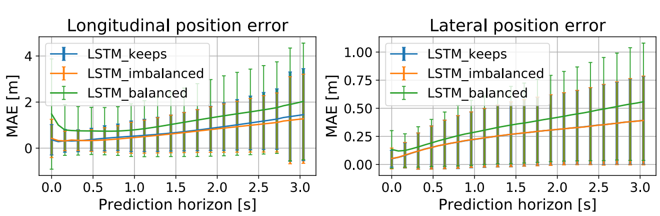


Fig. 7: MAE of the LSTMs trained on keeps, imbalanced and balanced data splits, and tested on keeps data split.

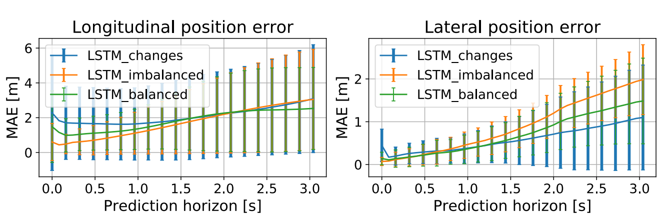


Fig. 8: MAE of the LSTMs trained on changes, imbalanced and balanced data splits, and tested on changes data split.

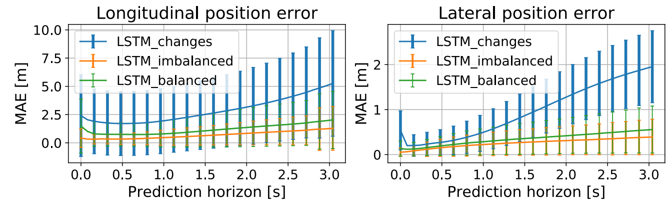


Fig. 9: MAE of the LSTMs trained on changes, imbalanced and balanced data splits, and tested on keeps data split.

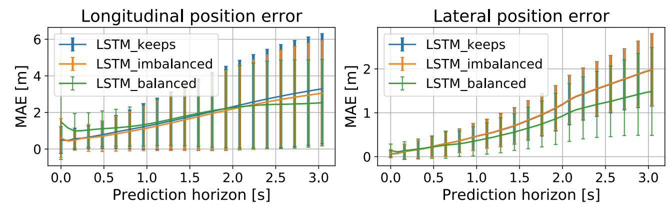


Fig. 10: MAE of the LSTMs trained on keeps, imbalanced and balanced data splits, and tested on changes data split.

trajectory during its training. However, the performance of the LSTM trained on *keeps* when predicting lane changes is not much worse than the LSTMs trained on *balanced* or *imbalanced* (Fig. 10).

If the target vehicle's intention was known a priori, to predict lane changes we would, depending on the desired N, use either the model trained on *balanced*, *imbalanced* or *changes*. Not having this knowledge of the target's intention, the LSTM trained on *balanced* would be a conservative choice, since although it does not yield the most accurate predictions, it does not suffer a severe performance decrease in case of misclassifications.

B. Hybrid Optimization

Since PB models perform well for $N < 1$, and DB models perform better for $N \geq 1$, we propose a hard constraining hybrid that combines the predictions of all its individual components with a weighted average. Given n prediction models, $M = \{m_1, m_2, ..., m_n\}$, the predictions of the hybrid model H for variable f at time t are given by

$$\hat{s}_{f,H}(t) = \sum_{m_i \in M} \hat{s}_{f,m_i}(t) \cdot w_{f,m_i,t}, \tag{11}$$

where the weight of each model's prediction for each variable at a given time, $w_{f,m_i,t}$, varies with N depending on the individual model's accuracy, and to find the weights for an optimal combination of the models, three approaches are compared: an exhaustive grid search (GS), a per-time step regression (Reg.), and a simple perceptron (Perc.), as shown in Fig. 11.

As the PB component, CV is considered. As the DB component, since there was no clear best model without having any knowledge about the target vehicle's intentions, all four implemented models are considered. Thus, $|M| = 5$, with one PB model and four DB models.

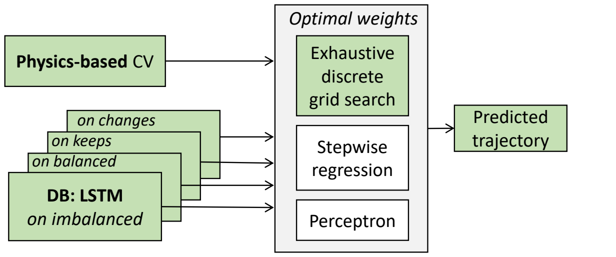


Fig. 11: Optimal selection of hybridization weights: one PB model and four DB models are combined with the best performing optimization method, the grid search.

For the optimization of the weights, all three methods use the training split of *balanced*, as to not favor either type of maneuver. Fig. 12 depicts $\epsilon_{x,\mu}$ and $\epsilon_{y,\mu}$ for $N \leq 1$ for all three approaches, which achieve very similar performance, but for longitudinal position prediction the grid search is marginally more accurate.

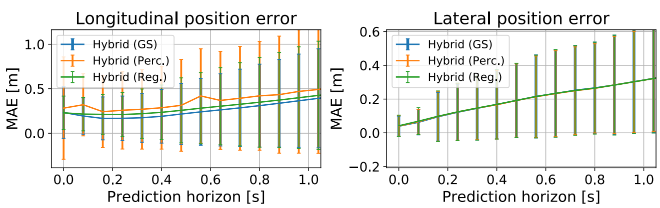


Fig. 12: Performance of the resulting hybrids using three methods to find the optimal weights: GS, Reg., and Perc.

Fig. 13 depicts the weights resulting from the grid search optimization. For longitudinal position, CV is consistently the model with the highest contribution. For lateral position, CV is only significant for a very short horizon, $N < 0.2$ seconds, and then it transitions to the LSTM trained on *keeps* and finally the one trained on *balanced*, while still being slightly influenced by the LSTM trained on *changes*. There seems to be no benefit of introducing the predictions of the LSTM trained on *imbalanced* for lateral prediction. Given no prior knowledge of the type of maneuver at hand, the hybrid optimized with the grid search performs better than any of its individual components, as shown in Fig. 14. However, when evaluating the performance of the hybrid separately on lane keeps and lane changes, it is not always more accurate than all of its components. For instance, when evaluated on *changes*, for $N \geq 1$, the LSTM trained on *changes* yields more accurate

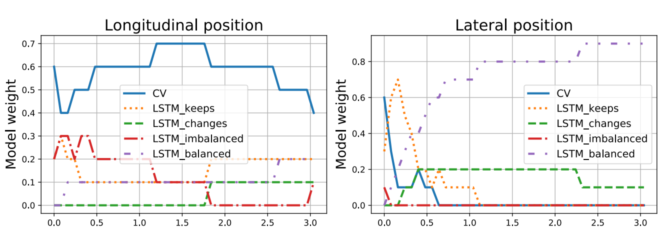
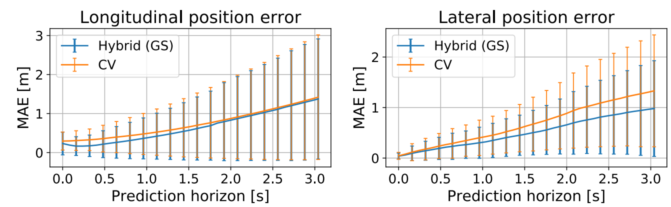


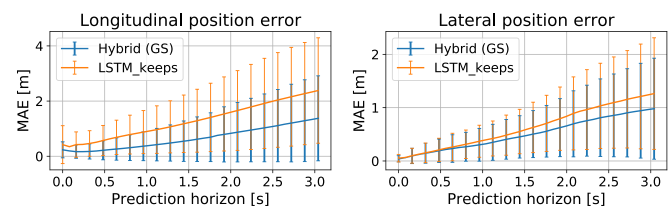
Fig. 13: Weights resulting from the grid search for each component of the hybrid model.

lateral position predictions. This difference in performance is expected, as *balanced* contains the same number of lane changing and lane keeping trajectories.

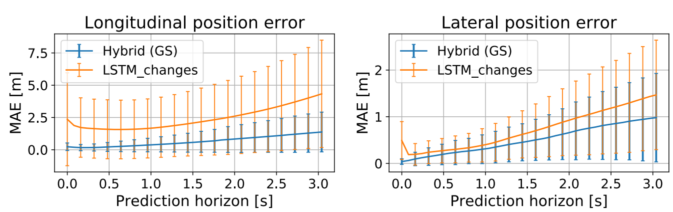
The accuracy of the models increases significantly if the type of maneuver is known a priori. However, this is an assumption we wish to avoid, therefore it remains a future research focus. Nevertheless, even without this assumption, the hybrid's predictions are highly accurate, having $\epsilon_{x,\mu} < 1$ meters for $N \leq 2.5$ seconds (see Fig. 14). Similarly, $\epsilon_{y,\mu} < 0.5$ meters for $N \leq 1.5$ seconds. This is a more challenging requirement and could be improved by using more, and optimally balanced training data.



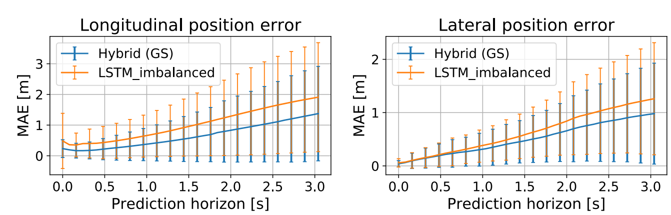
(a) Hybrid vs. CV



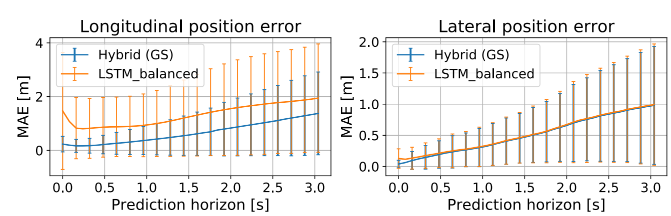
(b) Hybrid vs. LSTM trained on lane keeps



(c) Hybrid vs. LSTM trained on lane changes



(d) Hybrid vs. LSTM trained on the imbalanced split



(e) Hybrid vs. LSTM trained on the balanced set

Fig. 14: MAE of the proposed hybrid compared to each of its individual components on the balanced test set.

IV. Conclusions & Future Work

The contribution of this paper is two-fold. First, we introduce an optimal hybrid method for trajectory prediction of non-communicative on-road vehicles, and second we investigate in detail the performance and selection of the independent components that build this hybrid model.

We begin by performing a thorough analysis of the performance of different physics-based (PB) (evaluating two of the most commonly used models) and data-based (DB) models in predicting trajectories using on-road measured data. This analysis highlights the benefits of using PB models for short term prediction horizons and DB models for longer term. Furthermore, we investigate the impact of training our models on maneuver specific vs. mixed trajectories and we conclude that there is no clear best split of data to train this method if there is no prior information about the vehicle's intentions. We introduce safety performance indicators for lateral and longitudinal predictions and we show that the hybrid method achieves good accuracy within the safety bounds. This method is modular and easily extendable for predicting more on-road maneuvers.

Future work will address analyzing the optimal ratio of lane changes and lane keeps used during the training phase, and developing and comparing more complex hybrid models to improve the accuracy. Furthermore, to show the improvements in safety and comfort, the prediction models will be integrated with automated decision making.

References

[1] E. Van Nunen, J. Verhaegh *et al.*, "Robust model predictive cooperative adaptive cruise control subject to V2V impairments," in *IEEE Conference on Intelligent Transportation Systems*, 2018.

[2] J. Ploeg, A. F. A. Serrarens, and G. J. Heijenk, "Connect & Drive: design and evaluation of cooperative adaptive cruise control for congestion reduction," *Journal of Modern Transportation*, vol. 19, no. 3, p. 207–213, 2011.

[3] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," *Proceedings of the 11th Int. Conference on Information Fusion, FUSION 2008*, no. 1, 2008.

[4] F. Altche and A. De La Fortelle, "An LSTM network for highway trajectory prediction," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 353–359, 2018.

[5] G. Xie, H. Gao *et al.*, "Vehicle Trajectory Prediction by Integrating Physics- and Maneuver-Based Approaches Using Interactive Multiple Models," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5999–6008, 2018.

[6] M. Treiber, A. Kesting, and C. Thiemann, "How Much does Traffic Congestion Increase Fuel Consumption and Emissions? Applying a Fuel Consumption Model to the NGSIM Trajectory Data," *Transportation Research Board*, 2008.

[7] A. Elfar, C. Xavier, A. Talebpour, and H. S. Mahmassani, "Traffic shockwave detection in a connected environment using the speed distribution of individual vehicles," *Transportation Research Record*, vol. 2672, no. 20, p. 203–214, 2018.

[10] X. Lu and A. Skabardonis, "Freeway Traffic Shockwave Analysis: Exploring NGSIM Trajectory Data," *Transportation Research Board 86th Annual Meeting*, 2007.

[8] V. Punzo, M. T. Borzacchiello, and B. F. Ciuffo, "Estimation of Vehicle Trajectories from Observed Discrete Positions and Next-Generation Simulation Program (NGSIM) Data," *Transportation Research Board 88th Annual Meeting*, 2009.

[9] L. Leclercq, N. Chiabaut, J. Laval, and C. Buisson, "Relaxation phenomenon after lane changing experimental validation with NGSIM data set," *Transportation Research Record*, vol. 1999, no. 1, p. 79–85, 2007.

[11] B. Coifman and L. Li, "A critical evaluation of the Next Generation Simulation (NGSIM) vehicle trajectory dataset," *Transportation Research Part B: Methodological*, vol. 105, pp. 362–377, 2017.

[12] E. S. Gardner, "Exponential smoothing: The state of the art," *Journal of Forecasting*, vol. 22, no. 4, pp. 637–666, 1985.

[13] P. A. Moran and P. Whittle, "Hypothesis Testing in Time Series Analysis." *Journal of the Royal Statistical Society. Series A (General)*, vol. 114, no. 4, p. 579, 1951.

[14] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for R," *Journal of Statistical Software*, vol. 27, no. 3, 2008.

[15] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods," *Int. Journal of Forecasting*, vol. 18, no. 3, pp. 439–454, 2002.

[16] T. G. Dietterich, "Ensemble methods in machine learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2000.

[17] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," *In Practice*, 2006.

[18] F. Hamilton, A. L. Lloyd, and K. B. Flores, "Hybrid modeling and prediction of dynamical systems," *PLoS Computational Biology*, vol. 13, no. 7, pp. 1–20, 2017.

[19] A. Karpatne, W. Watkins, J. Read, and V. Kumar, "Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling," *arXiv preprint arXiv:1710.11431*, 2017. [Online]. Available: http://arxiv.org/abs/1710.11431

[20] X. Jia, J. Willard *et al.*, "Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles," *SIAM Int. Conference on Data Mining, SDM*, pp. 558–566, 2019.

[21] U. Forssell and P. Lindskog, "Combining Semi-Physical and Neural Network Modeling: An Example ofIts Usefulness," *IFAC Proceedings Volumes*, vol. 30, no. 11, pp. 767–770, 1997. [Online]. Available: http://dx.doi.org/10.1016/S1474-6670(17)42938-7

[22] T. Hirata, T. Kuremoto, M. Obayashi, S. Mabu, and K. Kobayashi, "Time Series Prediction Using DBN and ARIMA," in *Proceedings - 2015 Int. Conference on Computer Application Technologies, CCATS 2015*, 2016, pp. 24–29.

[23] P. G. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[24] O. Valenzuela, I. Rojas *et al.*, "Hybridization of intelligent techniques and ARIMA models for time series prediction," *Fuzzy Sets and Systems*, vol. 159, no. 7, pp. 821–845, 2008.

[25] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2664–2675, 2011.

[26] S. d. O. Domingos, J. F. L. de Oliveira, and P. S. G. de Mattos Neto, "An intelligent hybridization of ARIMA with machine learning models for time series forecasting," *Knowledge-Based Systems*, vol. 175, pp. 72–86, 2019. [Online]. Available: https://doi.org/10.1016/j.knosys.2019.03.011

[27] R. Gao, X. Guo *et al.*, "Aggressive driving saves more time? Multi-task Learning for Customized Travel Time Estimation," in *IJCAI Int. Joint Conference on Artificial Intelligence*, 2019.

[28] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014.

[29] N. Jouppi, C. Young *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings - Int. Symposium on Computer Architecture*, 2017.