Journal of Sustainable Development Indicators

http://www.sdewes.org/jsdi

Year 2026, Volume 2, Issue 4, 1030765

Original Research Article

Situation Aware Adaptive Online Driver Behaviour Prediction based on Vehicle Sensor Data

Tom Jäger*[*1], Steven Peters[1]
[1]Institute of Automotive Engineering
Technical University of Darmstadt
Otto-Berndt-Straße 2, Darmstadt, Germany
e-mail: tom.jaeger@tu-darmstadt.de

Cite as: Jäger, T., Peters, S., Situation Aware Adaptive Online Driver Behaviour Prediction based on Vehicle Sensor Data, J. sustain. dev. indic., 2(4), 1030765, 2026, DOI: https://doi.org/10.13044/j.sdi.d3.0765

ABSTRACT

Accurate energy consumption prediction and therefore velocity prediction, over mid-term is essential for efficient vehicle operation, particularly in electric and hybrid vehicles where range and resource management are critical. However, most current models fail to account for the real-time variability in driver behaviour, especially in response to dynamic traffic situations. This work proposes an adaptive system that identifies and adapts to behaviour patterns which are associated with repeatedly inaccurate predictions by comparing expected and actual driver actions. This allows the system to automatically learn from new behaviour patterns without the need for manual supervision. The proposed method is evaluated on exemplary driving situations extracted from open driving data. By fusing traffic context modelling with driver behaviour modelling, this method contributes to accurate velocity prediction systems, which can adapt to changes in driver behaviour over time.

KEYWORDS

Velocity Prediction, Behaviour prediction, Adaptive learning, Scene prediction, Driver behaviour, Speed prediction

INTRODUCTION

Energy efficiency is a critical factor in modern vehicle design, driven by the need to reduce operational costs and conserve resources. For electric vehicles (EVs), energy efficiency directly translates to increased range and reduced charging requirements, enhancing customer satisfaction. While hardware developments continue to improve vehicle efficiency, control strategies that optimize energy usage based on predicted system states are also a focus of recent research [1]. Employing such systems to increase energy efficiency increases vehicle performance in terms of driving range and contributes to broader sustainability objectives. In particular, it aligns with the United Nations Sustainability Goals (SDGs) [2], such as SDG 7 (Affordable and Clean Energy), by promoting efficient energy use and SDG 12 (Sustainable Consumption and Production) by reducing energy consumption without the need for additional natural resources.

Predictive control strategies, which depend on accurate forecasts of future system states, have gained attention as a promising means to enhance energy efficiency. Vehicle velocity is of special interest to model predictive control (MPC) strategies in vehicles, as it is important

[*] Corresponding author

for the state of many components and therefore also a necessary factor in predictive control [3]. MPC strategies for improved energy efficiency have been studied for many vehicle related areas, such as powertrain control in electric vehicles with multiple motors, where disconnect clutches in combination with MPC could achieve up to 6 % energy savings, on top up to 5 % savings achieved by torque distribution [4]. Similarly, an economic MPC strategy for HVAC control in electric vehicles has been shown to reduce energy consumption by 4.4 % to 7.5 % when compared to a rule-based approach [5]. Since powertrain energy use constitutes one of the largest contributors to a vehicle's overall energy usage, effective approaches for predicting vehicle velocity are of particular importance in MPC strategies for the powertrain. Gaikwad et al. [6] demonstrated a 2.1 % fuel saving potential in a hybrid vehicle employing a learning based velocity prediction and MPC for engine torque and engine speed control.

A variety of different approaches for velocity prediction have been studied, ranging from simple rule-based approaches to complex systems which incorporate a variety of input data. Simple kinematic models are generally limited to short prediction horizons, as they cannot capture complex driver behaviour and environmental interactions [7]. Current models showing the best performance typically employ machine learning to learn the interrelations of different datapoints and future vehicle velocity from collected driving data. Recent survey literature confirms that such data-driven and particularly deep learning based model are best suited to capture the variety and complexity of real world driving [8].

Driver behaviour plays an important role in shaping vehicle dynamics and, consequently, energy consumption [9]. Behaviour patterns directly influence the velocity trajectory, which in turn determines energy demand. This necessitates the incorporation of driver behaviour into prediction models for energy efficient predictive control. Even under similar traffic and road conditions, different drivers exhibit distinct tendencies in acceleration, braking, and following distance [10].

Driver behaviour is studied in a variety of contexts. In [11], a taxonomy of 200 behaviour modeling approaches is presented, showing that intention estimation, motion prediction and behaviour planning are of great research interest. Recent surveys in autonomous driving research highlight human behaviour as a key area of interest, emphasizing that understanding driver behaviour is crucial for enhancing trust and acceptance of intelligent driving systems [12]. However, most studies on driver behaviour either aim to model human behaviour explicitly or are based on labeled data collected from different drivers and aim to classify or predict driver behaviour based on models trained on this data. Such data is not publicly available on a large scale, necessitating significant data acquisition effort if behaviour should be incorporated in supervised learning methods.

The behaviour of human drivers is also context-dependent, adapting dynamically to roadway geometry, traffic conditions and interactions with other vehicles [13]. Additionally, driver behaviour exhibits variability, both across different drivers and over time, due to differences in individual characteristics and situational influences [14]. Many learning based velocity prediction systems do not adapt to such changes, which can lead to suboptimal prediction performance or drifting performance over time.

Online machine learning describes a class of methods which update models continuously as new data arrives. While classical machine learning approaches typically rely on large, fixed datasets for training and then apply the resulting model without modification, online methods are designed to adapt incrementally in response to streaming data. These methods have been shown to have practical benefits in the automotive context, for example in lane change behaviour prediction [15]. As online learning methods can help to mitigate some of the problems which are common with pretrained neural networks, like concept drift in time series forecasting [16], they are well suited to mitigate the influence of varying behavior in velocity prediction tasks.

To address the problem of heterogeneous driver behaviour in similar situations, a method is presented that integrates a pretrained velocity prediction model with an online learning system capable of adapting to the behaviour of individual drivers. The pretrained component establishes a strong baseline by leveraging large-scale datasets to capture general velocity patterns across diverse driving conditions. Building on this foundation, the online component incrementally refines predictions during runtime by compensating for systematic deviations associated with specific driving styles. By comparing the predicted velocity to the actual velocity, which can be measured after the prediction horizon passes, the system adjusts to changes in behaviour patterns. This process has the additional benefit that no large-scale labelled behaviour data is required for the supervised learning process, while large scale driving data from available datasets can be used.

The resulting framework combines the scalability and generalization capabilities of offline-trained models with the flexibility of online adaptation. In contrast to purely offline approaches, it can account for intra- and inter-driver variability, while avoiding the data inefficiency and instability often associated with fully online learning methods. This makes the approach well-suited for real-world applications, where large-scale datasets are available for pretraining, but driver-specific data is scarce.

The remainder of this paper is organized as follows: In the second section, the methodology is described, including the dataset, the baseline velocity prediction model and the online learning system. Section three presents the results for the baseline model, the selection of the scenarios for the evaluation of the online learning system and the evaluation of this system. Finally, the conclusion summarizes the key results, discusses remaining open questions, and outlines directions for future research and practical applications.

METHODS

This section describes the methods used in this work. First, the data source and preprocessing are introduced. The resulting data forms the basis for the training of the baseline velocity prediction model. Additionally, the sequence of scenarios that are relevant for the online learning system are extracted from this data. Afterwards, the architecture and training process of the baseline model are presented, followed by the design of the online learning system.

Driving Data

The source for the training and testing data for the studied prediction models is the nuPlan open dataset [17]. This dataset provides large-scale, high resolution driving data collected from urban environments in 4 cities, totalling over 1000 hours. The dataset contains high-frequency recordings of vehicle states, labelled sensor data, map information and annotated scenario information. From the dataset, the features listed in Table 1 where extracted. The velocity and acceleration features were computed from the directional velocities and accelerations available in the nuPlan dataset. The distance to the lead vehicle was computed from the absolute positions of the ego vehicle and the closest vehicle in the direction of travel. This distance to the leading vehicle was extracted from the labelled LiDAR objects lists nuPlan provides.

Table 1. Extracted Features

Feature	Explanation
v	Ego vehicle velocity, computed from directional data
a	Ego vehicle acceleration, computed from directional data
yaw	Ego vehicle yaw angle
angular_rate_x	Ego vehicle roll rate
angular_rate_y	Ego vehicle pitch rate
angular_rate_z	Ego vehicle yaw rate
dist_lead	Distance of ego vehicle to leading vehicle
type	Type of scenario

The data for the ego vehicle movement is available at a sampling rate of 100 Hz in the nuPlan dataset, while the scenario annotation is based on the environment and typically only available at a sampling rate of 20 Hz. To ensure a complete dataset, the empty rows in type column were filled with the type corresponding to the closest timestep. In almost all cases, this procedure fills in a scenario with a time offset of at most 30 ms, which is a good tolerance for the scenario type.

The resulting dataset was saved as a selection of 15 second sequences as the feature set for the machine learning models. The corresponding velocity 10 seconds after the last time step in each 15 second sequence was saved as the corresponding label for model training and evaluation.

Scenario Selection

To demonstrate the viability of the approach, scenarios with particularly high changes in vehicle velocity were extracted, as changes in vehicle velocity typically coincide with challenges in optimal control. The basis for the scenario classification is given by the scenario tags included in nuPlan. These tags include 75 different scenarios.

As the scenario tags are designed for fine-grained insights into automated driving systems, similar scenarios from a velocity prediction angle are sometimes categorized into different classes. To get a smaller set of classes, some scenario tags where combined to generate groups which are suitable for velocity prediction tasks. For acceleration scenarios, which are of special interest in the context of energy management tasks, distinct tags were combined. This includes for example the "acceleration at traffic light" and "acceleration at traffic light without lead" scenarios, which exhibit comparable velocity profiles.

To select suitable scenarios for the demonstration of the proposed approach, the scenarios were analysed based on their number of occurrences, and their average and maximal velocity changes over 10 s windows, as this corresponds to the prediction horizon of the used prediction model. For the evaluation of the online learning model, sequences of similar scenarios were extracted from individual trips. These trips can be obtained from the dataset by extracting the sequences with consecutive timesteps for the same vehicle. To include only trips with one driver, sequences were split when the vehicle remained stationary at the pickup and drop off locations for the respective city for longer than 5 seconds. While the dataset does not provide information about the driver, this way no driver change should appear within a sequence grouped as one trip. This method ensures transferability to the application of adapting to the driver behaviour in real world driving trips.

Velocity Prediction Model

The pretrained velocity prediction model is based on a long short-term memory (LSTM) architecture. This special form of recurrent neural network was originally proposed by Hochreiter and Schmidhuber [18], and has since been enhanced, notably by Gers et al. [19]. The input of the model consists of 15 second time windows of the driving data described in the driving data subsection, and the model is trained to predict the velocity of the ego vehicle after a 10 second time horizon. For training the model all scenarios in the dataset were used. This provides a realistic baseline comparable to other general purpose velocity prediction models.

For model training, hyperparameter optimization and testing, the preprocessed data was split into three parts after being shuffled randomly, creating a train, validation and test set containing 60 %, 20 % and 20 % of the samples respectively. These datasets were saved for reproducibility, ensuring that each model was trained on the same data during hyperparameter optimization.

The hyperparameters of the model were optimized using a Bayesian optimization process. In practice, this is achieved using the Optuna framework [20]. The optimized parameters include the sizes of the LSTM layers of the network, the dropout, the learning rate and the batch size. The ranges for the layer sizes have been chosen to encompass very compact models at 32 units per layer, up to rather complex models which are typically used for more complex parameter sets. Yeon et al. [21], while training on a different dataset, have shown good performance using layer sizes between 125 and 310 for a range of feature sets, which falls in the tested ranges in this work. The hyperparameter optimization has been performed, as exact network architectures are typically not directly transferrable to different feature sets and dataset compositions. An overview over the optimized parameters and the search space in given in Table 2.

Table 2: Hyperparameter search spaces for the Bayesian optimization

Hyperparameter	Search interval/set
LSTM units first layer	[32, 512]
LSTM units second layer	[32, 512]
Dropout	[0, 0.3]
Learning rate	[0.0001, 0.01]
Batch size	{16, 32, 64, 128}

During the Bayesian optimization, new candidates are chosen by balancing exploration of not yet tested parameters with exploitation of well-performing combinations. The model for each combination of tested hyperparameters was trained for 100 epochs on the train set, with the validation set used for validation during the training process. For the evaluation of the network, the mean absolute error (MAE) is used, which is calculated as the mean of the absolute value of all prediction errors in a sample set.

Online Learning System

To compensate for errors in specific situations which vary between trips, an online learning model is employed. The idea behind the system is to treat the difference between the baseline prediction and the true outcome as a separate learning problem. At each step, the baseline provides an initial prediction, and the online learner estimates the likely error of that prediction based on the current input features. For this the online learner obtains the same features as the baseline model, but only for the current time step, as well the prediction of the baseline model as input. Once the ground truth becomes available, the learner updates its internal model of the error, refining its understanding of how prediction errors vary with different feature patterns.

A simple online learning algorithm is applied to a selected set of scenarios to simplify the evaluation process. This simplification is justified, as the selected scenarios can be predicted with a high degree of accuracy, making the use of a simple online learning algorithm sufficient for evaluation. The sequences of these scenarios are fed into the prediction algorithm incrementally and the algorithm is reset after the scenarios which are extracted from one continuous trip have been used once. This ensures data which corresponds to the use case of adapting to individual driver behaviour, as driver behaviour is most consistent in individual trips. Additionally, sampling the data on trip-by-trip basis guarantees that the data is obtained from one driver and in a coherent set of conditions.

For the implementation of the online learning system, the River framework [22] was chosen as it is specifically designed for online machine learning and incremental updates. The mathematical model used by the online learner to capture the error pattern is based on linear regression. Given a feature vector x, the system computes the estimated error $\hat{e}$ using the weight vector w and bias b using a linear equation:

$$\hat{e} = w^T x + b \tag{1}$$

The weights w and the bias b are initialized as 0, ensuring that the model starts from a neutral baseline. After every prediction the true error, computed as the difference between predicted velocity and actual velocity, is obtained as the target for the online learner. The model parameters, i.e. the weights and the bias, are updated using the Adam optimizer with learning rate 0.05 provided by the River framework. During the optimization, an L2 regularization with a factor of 0.0005 punishes the system for large weights, which increases stability by preventing steep gradients of the linear function. Larger gradients could otherwise lead to large differences in predictions for very similar inputs.

RESULTS

The results are presented beginning with results of the hyperparameter optimization and the performance of the baseline velocity prediction based on the LSTM neural network. Afterwards, the baseline prediction error for the different scenario types present in the dataset is analysed and the scenarios on which the online learner is evaluated are selected. Finally, the online learning system is evaluated on sequences of the selected scenarios.

Baseline Velocity Prediction

After optimizing the hyperparameters of the baseline velocity prediction model, the result is a model with two hidden layers with 146 and 131 nodes respectively. The learning rate was set to 0.0004, the batch size to 64 and the dropout was set to 0.2. The losses of the 100-epoch training process of this model are illustrated in Figure 1. Note that the loss is calculated on the normalized data and does not represent the error in terms of meters per second.

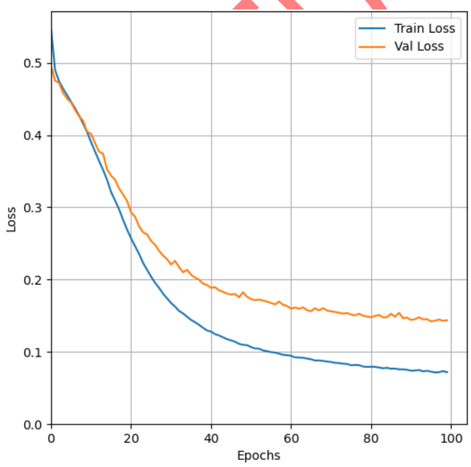

Figure 1: Training losses of the baseline model

The LSTM model trained on the dataset after the hyperparameter optimization has a MAE of 0.87 when evaluated on the test dataset. Figure 2 shows the distribution of the first 500 samples of the test set, highlighting the range and variability of the target values encountered during evaluation. The results indicate that the model is overall well suited for the velocity prediction task, as most predictions are concentrated around the diagonal. However, some points show a clear variability in prediction estimates, indicating potential for improved prediction accuracy. While such dispersions appear at all velocities covered in the dataset, they are relatively more common at low speeds, stemming from prediction difficulties in acceleration and deceleration scenarios.

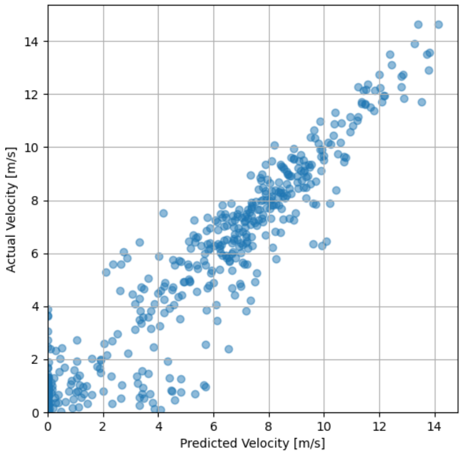

Figure 2: Actual vs predicted velocities of the baseline model

Scenario Analysis and Sequence Extraction

The prediction error of the baseline model varies noticeably between different driving scenarios. In general, situations that involve larger deviations between the initial velocity at the start of the prediction horizon and the velocity observed at the end of that horizon tend to produce higher errors. By contrast, scenarios in which the vehicle maintains a speed closer to a constant driving velocity result in considerably smaller prediction errors. In these cases, the model's forecasts align more closely with the actual behaviour of the system, leading to improved accuracy. An overview of the three scenario types with the lowest average MAE and the three scenario types with the highest average MAE is given in Table 3.

Table 3: MAE of example scenario types

Scenario type	MAE [m/s]
following_lane_with_slow_lead	0.49
near_multiple_vehicles	0.52
medium_magnitude_speed	0.54
accelerating_at_traffic_light	1.15
Stopping_at_stop_sign_without_lead	1.18
accelerating_at_crosswalk	1.18

As scenarios with higher changes in velocity are more interesting from the perspective of energy management and exhibit larger errors by the baseline model, acceleration scenarios form the basis for the evaluation of the proposed system. In particular, the scenarios used for the extraction of sequences from continuous trips are listed in Table 4. These scenarios encompass a variety of acceleration scenarios which are reliably predictable by a scenario classification system.

Table 4: Selected scenario types

accelerating_at_stop_sign
accelerating_at_stop_sign_no_crosswalk
accelerating_at_crosswalk
accelerating_at_traffic_light

Based on these categories, 289 sequences of 10 or more acceleration scenarios that are part of the test set could be extracted. Each of these sequences consists only of scenarios extracted from on continuous trip. These sequences represent a diverse set of driving situations in which the vehicle transitions from a stationary or low-speed state to higher velocities.

Online Learning System

On sequences of scenes with high error and consistency of the error in terms of the baseline model either mostly overestimating or underestimating the future velocity, the online learning algorithm led to the most significant decrease in prediction error. Of the studied sequences of acceleration scenarios, the largest error reduction reduced the MAE calculated over the predictions in the sequence from 1.12 to 0.5, representing an error reduction of 55 %. This reduction was achieved on a 16-sample sequence of scenes with positive prediction errors, visualized in Figure 3.

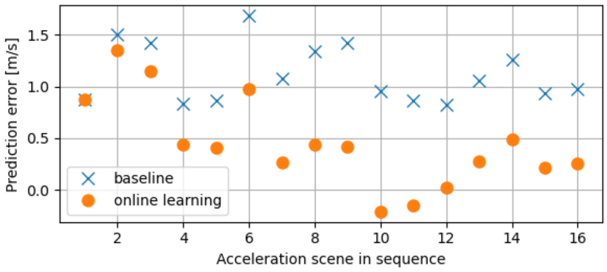

Figure 3: Error reduction on sequence with positive errors

A comparable performance can be observed on sequences in which the prediction error of the baseline model is uniformly negative. The largest MAE in such a sequence was achieved on a 15-sample sequence, which is visualized in Figure 4. In this sequence, the online learning system reduced the MAE from 1.17 to 0.56, representing a 52 % error reduction, further demonstrating the systems capability to adapt to uniform error patterns.

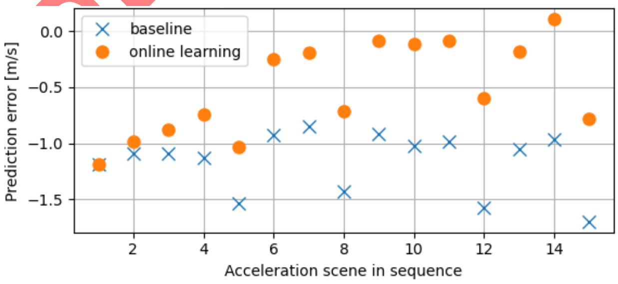

Figure 4: Error reduction on sequence with negative errors

While the illustrated sequences represent the best case for the online learning model, the assessment of the approach on the full set of sequences shows an average MAE reduction of 0.17, which represents a 15 % error reduction on the evaluated scenes. Overall, the method leads to a reduction in MAE on 84 % of the evaluated sequences. The histogram in Figure 5 shows the MAE reduction over all sequences by percentiles, showing that in most cases, the system improves the velocity prediction.

Year 2026
Volume 2, Issue 4, 1030765

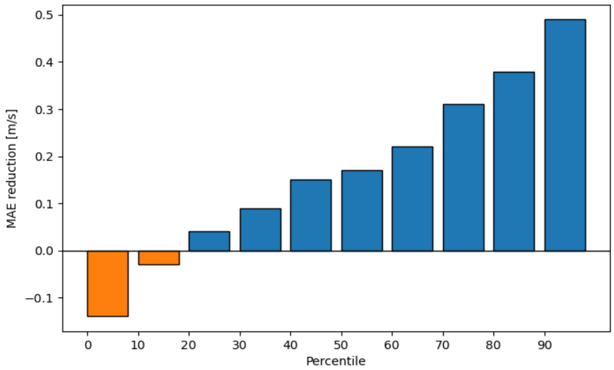

Figure 5: MAE reduction through online learning by percentile

No reduction or even an increase of the prediction error through the online learning system occurs mostly in sequences in which there is an alternating pattern of prediction errors produced by the baseline system. In such cases, the online learning system cannot learn a clear trend in the data which leads to only small changes of the initial prediction. This might increase the prediction error. Figure 6 shows an example of an alternating error sequence. In the corresponding sequence of scenarios, the overall MAE was increased by 0.08 m/s, placing it in the lowest percentile of sequences as ranked by the performance of the online adaptor. Sequences of predictions with alternating errors cause the model to adapt poorly, because the learned linear regression can only fit the error patterns with a linear function and the regularization prevents steep gradients. While this increases stability in general, in this case it prevents the model from learning markedly different errors for situations with similar features.

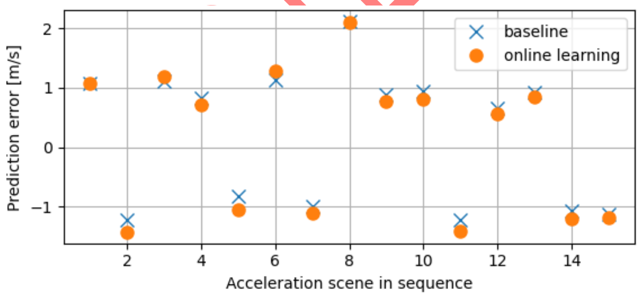

Figure 6: Online learning performance in mixed error sequence

It is important to note that, while the possible performance benefits of the presented method in sequences with favourable conditions, usually expressed as uniform errors of the baseline model, are larger than the losses under poor conditions. One reason for this lies in the fact that alternating errors tend to decrease the impact of the online learning system. In such cases, the impact and therefore the error increase caused by the adaptor is low.

CONCLUSION

The evaluation of the online learning system on acceleration scenarios demonstrates that shifts in driver behaviour relative to a baseline model can be effectively captured during runtime, without the need for large amounts of labeled data covering a wide range of driving styles. Since individual drivers tend to maintain consistent behaviour within a trip, but behaviour patterns differ substantially across drivers, these results highlight that velocity prediction models can be adapted on-the-fly to the individual characteristics of a driver by means of a lightweight online adjustment of model outputs, combined with a scene detection based on ADAS sensor data. While this study has so far only examined acceleration scenarios, the results show clear benefits of the proposed approach.

Across the complete set of extracted sequences, the method achieves an average mean absolute error reduction of 0.17, corresponding to a relative error reduction of approximately 15 % on the evaluated scenes. Moreover, 84 % of the sequences show an improvement over the baseline, which indicates that the system provides a robust and consistent benefit. The distribution of improvements, as illustrated in Figure 4, further underlines that most cases profit from online adaptation. The largest benefits are achieved on sequences that exhibit a uniform over- or underestimation by the baseline model.

When considering the real-world applicability of the proposed approach, its current empirical scope should be interpreted with caution. Because the system was evaluated only in acceleration scenarios, the reported findings cannot yet be assumed to generalize to all other possible driving scenarios. Extending the evaluation to a variety of contexts is therefore not merely a matter of completeness, but a necessary step toward demonstrating robustness under realistic operating conditions. Additionally, while care was taken to ensure that continuous trips were extracted and thus likely correspond to consistent driving behaviour, the dataset does not provide explicit driver identifiers. Consequently, the observed adaptations may not exclusively reflect individual driving styles but could also be influenced by other factors such as route characteristic or traffic conditions, which limits the strength of conclusions regarding true personalization.

A related methodological issue concerns the choice of online learner. The present linear regression model offers simplicity, interpretability, and low computational cost, but it is structurally limited when driver behaviour depends on nonlinear, stochastic, or interaction-rich patterns. This motivates a systematic comparison with nonlinear or hybrid alternatives, ideally under explicit constraints on stability, calibration effort, and runtime feasibility. Such an extension becomes even more relevant because the current personalization mechanism is trained on only one scenario type. A broader use of available data could not only accelerate adaptation to individual drivers but also make it possible to study whether driver-specific adjustments learned in one context transfer to others. This transfer question is particularly important for longer prediction horizons, which are highly relevant for energy management and predictive control but become increasingly difficult as uncertainty grows with both horizon length and scenario complexity.

Finally, the practical value of the achieved velocity prediction accuracy remains only partially understood. Prior work clearly shows why speed forecasts matter for eco-driving and predictive energy management, yet the downstream benefit depends on how prediction errors propagate through the subsequent control and optimization layers. Future studies should therefore complement statistical prediction metrics with task-level criteria such as energy consumption, control quality, and closed-loop robustness.

ACKNOWLEDGEMENT

This work was funded by the European Union's HORIZON EUROPE research and innovation program through the MINDED project under grant agreement No 101138201. The content of this paper is the sole responsibility of the author and does not necessarily represent the view of the European commission or its services.

NOMENCLATURE

Symbols

a	acceleration	[m/s^2]
b	bias	
$\hat{e}$	predicted error	
v	velocity	[m/s]

w	weights vector
x	features vector

Abbreviations

ADAS	Advanced Driver Assistance System
EV	Electric Vehicle
MPC	Model Predictive Control
LiDAR	Light Detection and Ranging
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error

REFERENCES

Journals

1. D. T. Machacek, S. van Dooren, T. Huber, and C. H. Onder, "Learning-Based Model Predictive Control for the Energy Management of Hybrid Electric Vehicles Including Driving Mode Decisions," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 5113–5127, Apr. 2024, https://doi.org/10.1109/TVT.2023.3333250.
2. United Nations General Assembly, "Transforming our world: The 2030 Agenda for Sustainable Development," United Nations, New York, Resolution A/RES/70/1, 2015. [Online]. Available: https://sdgs.un.org/goals.
3. A. Rezaei and J. B. Burl, "Prediction of Vehicle Velocity for Model Predictive Control," *IFAC-Pap.*, vol. 48, no. 15, pp. 257–262, 2015, https://doi.org/10.1016/j.ifacol.2015.10.037.
4. B. Škugor, J. Deur, W. Chen, Y. Zhang, and E. Dai, "Optimization of straight-line driving torque vectoring for energy-efficient operation of electric vehicles with multiple motors and disconnect clutches," *Optim. Eng.*, vol. 26, no. 2, pp. 753–780, July 2024, https://doi.org/10.1007/s11081-024-09902-7.
5. K. H. Kwak, J. Kim, Y. Chen, Y. Kim, and D. Jung, "Economic Model Predictive Control of HVAC System in Electric Vehicles," *IFAC-Pap.*, vol. 54, no. 20, pp. 852–857, 2021, https://doi.org/10.1016/j.ifacol.2021.11.278.
6. T. D. Gaikwad, Z. D. Asher, K. Liu, M. Huang, and I. Kolmanovsky, "Vehicle Velocity Prediction and Energy Management Strategy Part 2: Integration of Machine Learning Vehicle Velocity Prediction with Optimal Energy Management to Improve Fuel Economy," presented at the WCX SAE World Congress Experience, Apr. 2019, https://doi.org/10.4271/2019-01-1212.
7. S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, p. 1, Dec. 2014, https://doi.org/10.1186/s40648-014-0001-z.
8. T. Wang, D. Xiao, X. Xu, and Q. Yuan, "A Survey on Vehicle Trajectory Prediction Procedures for Intelligent Driving," *Sensors*, vol. 25, no. 16, Aug. 2025, https://doi.org/10.3390/s25165129.
9. M. Jaafar, H. Kadi, J. Z. Flores, S. Moosavi, and J. Pierre Radoux, "Driver behavior study for better speed prediction: unique use of speed sensor and machine learning," *Transp. Res. Procedia*, vol. 72, pp. 1645–1652, 2023, https://doi.org/10.1016/j.trpro.2023.11.636.
10. A. Laureshyn, K. Åström, and K. Brundell-Freij, "From Speed Profile Data To Analysis Of Behaviour," *IATSS Res.*, vol. 33, no. 2, pp. 88–98, 2009, doi: 10.1016/S0386-1112(14)60247-8.
11. R. Bhattacharyya, K. J. Brown, J. Wang, K. Driggs-Campbell, and M. J. Kochenderfer, "A Taxonomy and Review of Algorithms for Modeling and Predicting Human Driver Behavior," *Proc. IEEE*, pp. 1–23, 2025, https://doi.org/10.1109/JPROC.2025.3617487.

12. X. Liao *et al.*, "A Review of Personalization in Driving Behavior: Dataset, Modeling, and Validation," *IEEE Trans. Intell. Veh.*, vol. 10, no. 2, pp. 1241–1262, Feb. 2025, https://doi.org/10.1109/TIV.2024.3425647.
13. M. Treiber and A. Kesting, "General Aspects," in *Traffic Flow Dynamics: Data, Models and Simulation*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 55–59.
14. S. Ossen and S. P. Hoogendoorn, "Heterogeneity in car-following behavior: Theory and empirics," *Transp. Res. Part C Emerg. Technol.*, vol. 19, no. 2, pp. 182–195, Apr. 2011, https://doi.org/10.1016/j.trc.2010.05.006.
15. X. Liao *et al.*, "Driver Digital Twin for Online Prediction of Personalized Lane-Change Behavior," *IEEE Internet Things J.*, vol. 10, no. 15, pp. 13235–13246, Aug. 2023, https://doi.org/10.1109/JIOT.2023.3262484.
16. L. Zhao and Y. Shen, "Proactive Model Adaptation Against Concept Drift for Online Time Series Forecasting," 2025, https://doi.org/10.1145/3690624.3709210.
17. H. Caesar *et al.*, "NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles." arXiv, 2021, https://doi.org/10.48550/ARXIV.2106.11810.
18. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, https://doi.org/10.1162/neco.1997.9.8.1735.
19. F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000, https://doi.org/10.1162/089976600300015015.
20. T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, July 2019, pp. 2623–2631, https://doi.org/10.1145/3292500.3330701.
21. K. Yeon, K. Min, J. Shin, M. Sunwoo, and M. Han, "Ego-Vehicle Speed Prediction Using a Long Short-Term Memory Based Recurrent Neural Network," *Int. J. Automot. Technol.*, vol. 20, no. 4, pp. 713–722, Aug. 2019, https://doi.org/10.1007/s12239-019-0067-y.
22. Jacob Montiel *et al.*, "River: Machine learning for streaming data in Python," *J. Mach. Learn. Res.*, vol. 22, Apr. 2021, [Online]. Available: https://www.jmlr.org/papers/volume22/20-1380/20-1380.pdf.