Exploiting Maneuver Dependency for Personalization of Driver Assistance Systems

Hien Dang* and Johannes Fürnkranz*

Zusammenfassung:
The problem of personalization in Advanced Driver-Assistance Systems (ADAS) has been addressed in different studies. In general personalization is an approach which aims to develop a system that can adapt on the driver's behavior and thus improve the prediction performance. The individuality of such a system can be used to improve driving experience and comfort. High prediction performance also means that the model can better foresee driver's actions and accordingly raises a warning in case of hazard. In this work, we investigate in the dependency between different maneuvers. We propose an approach to extract information from past maneuver executions and use it as input for predicting impending maneuvers. In particular, we apply our method to predict which gap will be taken at a left-turn scenario where the driver has to wait for an appropriate gap before turning. The results show that by incorporating the previous maneuver execution the prediction performance can be increased by more than 9 % in term of F1 score. By running our approach using different types of past maneuver for a specific application, we can compare the amount of individual information of drivers contained in each type of maneuver.

Schlüsselwörter: Personalization, LSTM, Maneuver Dependency

1 Introduction

It is often the case that users have to adapt themselves to a new system or function to be able to use it. When the users' expectation and preference of a system are not met, their trust in the system will decrease and eventually they may ignore or turn it off . For a standard not-adapting system, there could be a gap between the user's expectation and the outcome of the system. Personalization systems aim at closing this gap.

The problem of personalization is already addressed in different field studies and applications. Recommender systems are an example, in which personalization play a central role. In recommender systems, the data are mostly sequence of actions [1] (i.e. "visiting a website" or "buying an item"...) The problem of personalization in this case is usually formulated as a prediction task of the next action A_i given the past action A_{i-1} or a sequence of the past actions $A_{0...i-1}$:

$$\pi : A_0...A_{i-1} \rightarrow A_i \tag{1}$$

*The authors are with Knowledge Engineering Group, TU Darmstadt, Hochschulstraße 10, D-64289 Darmstadt. (e-mails: {hiendang, fuernkranz}@ke.tu-darmstadt.de).

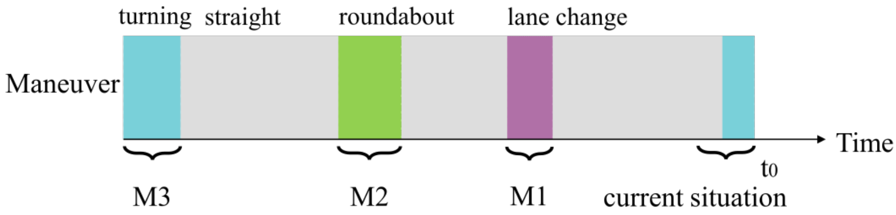


Figure 1: Representation of a test route as consecutive driving maneuvers

In the automotive field, the sequence of maneuvers may by itself reveal very little about the driver. For example, if a driver wants to get to his travel destination, certain maneuvers have to be performed in any case (i.e. entering a roundabout, turning at an intersection...). However, different drivers vary in the way they perform these maneuvers. As a consequence, the problem of personalization must in this case be based on observing how a driver performs the driving maneuvers to infer the driver's preference or to predict how s/he would perform the next maneuver or some specific maneuver of interest. Thus we need to learn a function which maps the driving style $f(.)$ of previous maneuvers to the driving style of an upcoming maneuver:

$$\pi : f(A_0)...f(A_{i-1}) \rightarrow f(A_i) \tag{2}$$

Understanding the driver is a challenging task, there are many factors that can influence a driver on her decision making process. These influencing factors include the driver's habit, her driving skills, her experience, her current physical and mental state and so on. Observing all these factors are complex, and modeling how they will affect the driver and the way s/he drives is also not easy. In this work, we consider a route as a sequence of maneuvers, interleaved with periods of default activities such as driving straight (see Fig 1), and approach the problem of personalization by learning the dependency between maneuver executions. We formalize the problem and show that it is reasonable to exploit the dependency between maneuver executions to adapt the model to individual drivers.

2 Related Work

The idea of personalization is to enable the system to adapt itself to the driver and thus is a key to further improve the safety and driving comfort. The advantages of personalization have already been observed in different applications in automotive field, such as adaptive cruise control (longitudinal driving assistance) or lane changing prediction (lateral driving assistance). Personalization is often combined with the idea of driving style recognition, since both problems address the same issue of understanding the driver behavior.

One of most common approaches is to divide drivers in different groups (e.g., aggressive, moderate and calm). The number of groups varies from two to six, depending on the particular application of interest. The group information could either be used directly to assign individual recommendation to driver or it could also be used as input to the personalization of some specific functionality of ADAS. In [2], Rosenfeld et al. adapt the driver model of [3] to divide drivers into three groups, and use this information together with demographic features for predicting the driver's preferences in the use of adaptive cruise control (ACC). The driver groups in this study are determined by rules that are manually defined based on empirical observations in the dataset. In contrast to this, Constantinescu et al. [4] present a data-driven approach which makes use of hierarchical clustering and principal component analysis (PCA) to discover six different driver groups. This approach does not make assumptions about driving styles but separates the driver groups by exploring and finding the intrinsic structures of the dataset. Since this method is unsupervised, the discovered clusters do not have an explicit meaning but one can assign suitable labels to each cluster by looking at the features that describe them.

Another approach for identifying driving style is collect and analyze self-reports from drivers [5, 6]. The drivers are asked to complete a driving style questionnaire, based on which different driver groups are formed. This kind of approach, however, suffers from subjective perceptions about driving style. For example, the perception of aggressiveness or calmness may vary from driver to driver.

In many applications in the automotive field, the personalization problem can be reduced to identifying the individual gap acceptance of the driver. In [7], Butakov et al. propose an approach to personalize the lane change prediction by modeling the longitudinal adjustment behavior and the gap acceptance of driver. The gap at a lane change maneuver is defined as the space between the leading vehicle and the following vehicle at the target lane. For adaptive cruise control, an adaptive system will automatically adjust the distance to the leading vehicle to match the driver's preference. Another application of gap acceptance is to predict the decision of the driver at an intersection where driver has to wait for a appropriate gap to perform a turning maneuver [8].

3 Approach on Personalization

Driving style is an abstract concept that is not directly observable from the data. What we can observe is the reflection of driving style on some measurable signals like speed, acceleration, etc. A survey conducted by Martinez et al. [9] shows that the choice of input signals to use for detecting driving style variates from application to application. In general the input signals could be divided into three main groups: vehicle dynamics, energy consumption and personality traits. In this work we focus on the first category of input signal since it is more general and always available in all maneuver execution.

3.1 Problem Formulation

Maneuver Execution Given a situation S_i and a driver D_j, the driving maneuver at this situation could be formulated as a function of S_i and D_j

$$M_i = f(S_i, D_j + \epsilon_i) \tag{3}$$

where S_i represents all environmental factors, such as traffic situation or weather, D_j constitutes the driver's dispositional factors. Both S_i and D_j have an impact on how a maneuver will be performed. However, the same driver could behave differently even if the same situation repeats at some other time. Thus, a variable ϵ_i is introduced to capture the fluctuations in the driver's behavior. It can be seen as a noise term of driver D_j while performing a maneuver M_i.

Personalization of a Driver Model Given a driver D_j at a situation S_i, we now want to predict a specific target value y_i for the current situation. y_i could be, for example, the next action of the driver. To personalize the prediction of y_i, we additionally have to consider the influences of the driver $D_j + \epsilon_i$ in this situation, since y_i depends on these both factors:

$$y_i \leftarrow S_i, D_j + \epsilon_i \tag{4}$$

Now we have the same problem as in modeling maneuver execution, namely that the driver's dispositional factors, which influence y_i, are not directly observable. This make it impossible to learn the individual impacts of driver D_j on y_i. The problem is usually reduced to predict y_i given the current situation S_i. This approach comes with an assumption that the individual influence from a driver on y_i is small or the behavior of the drivers in training set is similar to testing set.

Incorporating Previous Maneuvers As mentioned above, in real-world driving data, the drivers' factors are encoded in each maneuver execution M_i. We therefore propose that instead of learning y_i as a function of S_i and D_j, we can learn y_i as a function of S_i and M_{i-1}, where M_{i-1} is a maneuver that was recently performed by the same driver D_j. This temporal restriction of M_{i-1} allows us to approximate the current impact of the driver $D_j + \epsilon_i$ with $D_j + \epsilon_{i-1}$, which was captured in the previous maneuver.

$$\begin{aligned} y_i &\leftarrow S_i, M_{i-1} \\ &\leftarrow S_i, f(D_j + \epsilon_{i-1}, S_{i-1}) \\ &\approx S_i, f(D_j + \epsilon_i, S_{i-1}) \end{aligned} \tag{5}$$

The prediction of y_i can then be written as a function G of S_i, $D_j + \epsilon_i$ and S_{i-1}:

$$\begin{aligned} y_i &= g(S_i, f(D_j + \epsilon_i, S_{i-1})) \\ &= G(S_i, D_j + \epsilon_i, S_{i-1}) \end{aligned} \tag{6}$$

By taking M_{i-1} into account, we are also using the previous traffic situation S_{i-1} for predicting the current situation. With the assumption that y_i does not depend on other situations than the current S_{i-1}, there should be no information of y_i contained in S_{i-1}. G will be forced to learn to extract useful information about the driver from past maneuvers and then how they will affect the decision of the driver in the current situation.

3.2 Extracting Driver's Information

A maneuver execution (M_i) is characterized by a multivariate time series of sensor values:

$$M_i = (x_{i,1}, x_{i,2}, ..., x_{i,n}), \quad n > 0 \tag{7}$$

where n is the length of the time series and $x_{i,t}$ is a column vector of sensor values at time t. It has to be noted that the length of a maneuver varies, depending on the driver and the traffic condition. To learn from this kind of data we either have to convert them into fixed-length feature vectors or make use of a models that can handle arbitrary length.

In this work, we used two approaches to extract a driver's information from the last executed maneuvers. The first approach extracts features from the time series by computing the statistical information (i.e. minimum, maximum and standard deviation) from

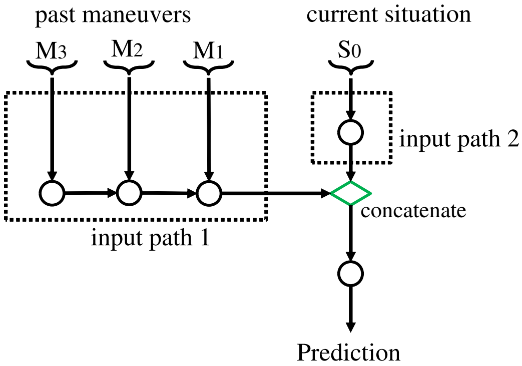


Figure 2: Proposed architecture to capture the driver information from previous maneuver executions and use that for predicting the current situation

each sensor. The second approach makes use of recurrent network layers and takes the whole maneuver execution as an input. While the idea of the first approach is widely used in the literature for extracting driver's information, the second one does not make any assumption about the statistical values but learns to extract useful information direct from raw data.

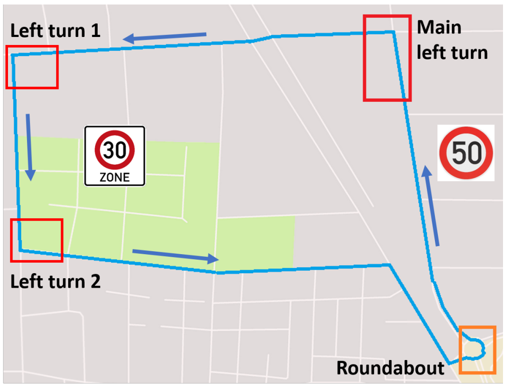
4 Modeling Maneuver Dependency

We have now formulated the prediction of y_i as a function of the current situation S_i and the previous maneuver M_{i-1}. By incorporating the previous maneuver M_{i-1} as input, the individual influence of the current driver will also be considered for predicting y_i. This can be extended by incorporating k last maneuvers that were performed by this driver $(M_{i-1}, M_{i-2}...M_{i-k})$. Fig. 2 shows an example of a network that additionally uses $k = 3$ last maneuvers as features. To validate this concept we test it with $k = 1$ which is theoretically easier to train and requires less data. We use a neural network to model the function G, whose objective is to predict the taken gaps.

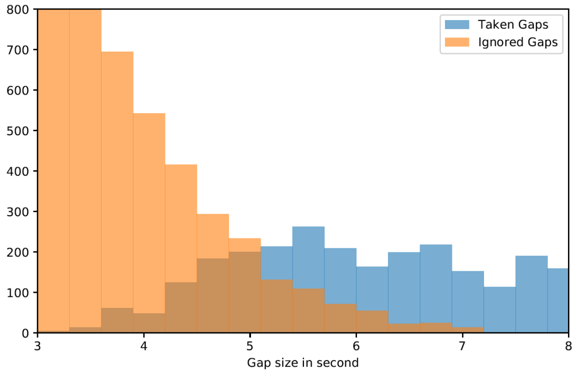
4.1 Network Architectures

We design the network using two input layers separately, one takes the current situation (S_i) as input and the other is used for capturing useful information from the previous maneuver (M_{i-1}). Each of these two input layers are then followed by hidden layers and form two separate paths. For the ultimate purpose of predicting y_i, these two paths are then combined in the deeper layers, followed by further hidden layers and lastly the output layer. The whole network is trained by back-propagating the classification error.

Using this network structure allows us to customize the two input paths individually and make it possible to apply different regularization strategies on each path. This is useful to deal with overfitting since it could be a problem when using previous maneuvers. Furthermore, the input path for capturing past maneuver M_{i-1} could be removed from the architecture, which results the common approach that learn to predict y_i directly from S_i.



(a) Test route for recording data. The roundabout and left turn are marked in orange and red



(b) Histograms of taken and ignored gaps

Figure 3: Visualization of driving route and statistics of gaps at the left-turn situation

4.2 Extracting Driver Information with Neural Network

As mentioned in 3.2, extracting driver information from M_{i-1} could be done in two different ways. We modeled the first approach by configuring the first input path using fully connected layers that take statistical values from the last maneuver execution as input. The function G could then be written as $G(S_i, h_s(M_{i-1}))$ or $G_s(S_i, M_{i-1})$ for short. Here h_s compute the statistic information of the given maneuver. The second approach makes use of a recurrent layer to capture the maneuver execution. In particular, we use Long short-term memory networks (LSTM) [10] for capturing past maneuvers M_{i-1}. LSTM Networkss have proven to be quite successful for sequence learning problems [11,12]. The function G in this case is formalized as $G(S_i, h_{lstm}(M_{i-1}))$ or $G_{lstm}(S_i, M_{i-1})$ for short. Here h_{lstm} depicts the LSTM layer that return hidden representation of the given maneuver M_{i-1}.

By using different type of past maneuvers, the performance of the trained model can be used for measuring which maneuver contains more useful information about the driver. Such maneuvers can then be used to characterize the driver and serve as input for further adaptive systems.

5 Experiments

Data set The data used in this work were collected from 32 drivers which cover a wide range of ages and driving experiences. From each driver we recorded 30 rounds of driving on a pre-defined route. The data set was collected in real-world traffic (see Fig 3a). The chosen route is a common urban route which requires the driver to perform different maneuvers like roundabout, left turn at 50 km/h zone (main left turn) , intersection with left yields to right, left turn at 30 km/h zone (left turn 1 and 2), etc. For our experiments, we focused on the two most complex maneuvers: roundabout and left-turn at 50 km/h zone.

For capturing the driving style we use four signals as features for describing a maneuver execution, which include speed, longitude acceleration, latitude acceleration and steering wheel speed. As features for describing the intersection situation we additionally use the

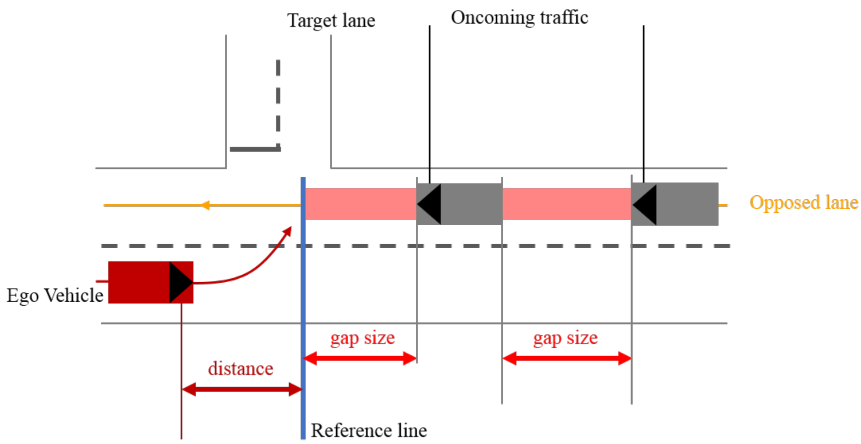


Figure 4: Illustration of the left turn situation with two possible gaps to take

position of the ego vehicle to determine the distance to the middle of the intersection.

Gap Acceptance at Left-Turn Maneuver In an unsignalized intersection scenario, in which the incoming traffic has the right of way. We observe that the preference of the driver in choosing gaps differs from driver to driver. Gaps that are in range from approximately 3 to 7 seconds could either be taken or ignored, depending on the driver's preference, as shown in Fig. 3b. In such a scenario, the problem of predicting the gap acceptance of driver should not only consider the current traffic situation but also the driver's individual preferences.

We detect the oncoming traffic using the equipped front-radar: their the relative position, speed and size. Based this information, each potential gap for turning left is computed. An illustration of the application at an intersection is shown in Fig. 4, where the ego vehicle is depicted as a red box and the incoming vehicles are shown in gray. The size of the first gap is the distance between the closest incoming vehicle and the middle of the intersection (reference line). The last gap is computed based on the furthest detected incoming vehicle and the maximum radar range. All gaps between these two gaps are bounded by a leading vehicle and a following vehicle. For each gap we compute the time (t_i) that is available for the driver to turn left: $t_i = s_i/v_i$, where the s_i is the longitudinal size of gap i and v_i is the current speed of the following vehicle that forms gap i.

6 Evaluation

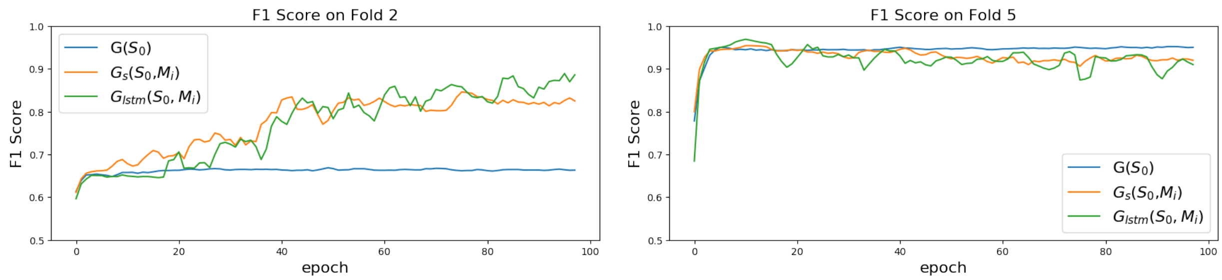
6.1 Evaluation Setup

We evaluate our method using 10-fold cross-validation to construct training and testing sets. The data are split into 10 equal and disjoint folds according to the driver ID. This assures that the model will be evaluated unseen drivers. In total we evaluate four models. As the baseline model, we estimate the best possible threshold that separates the taken gaps from the ignored gaps in the training set. The baseline is then compared to the three models described in the previous sections:

- $G(S_i)$ only use the current situation as input (the first input path is removed).

Table 1: Mean of $F1$ and accuracy score on validation set over 10-fold cross-validation

Model	F1 Score	Accuracy
Threshold	81.9 %	89.8 %
G(S_i)	86.8 %	92.6 %
$G_s(S_i, M_{i-1})$	88.8 %	93.2 %
$G_{lstm}(S_i, M_{i-1})$	91.1 %	94.7 %



(a) Incorporating M_i lead to improvement

(b) Overfitting on training data

Figure 5: Learning curve on different validation folds

- $G_s(S_i, M_{i-1})$ uses the statistical values from the roundabout maneuver
- $G_{lstm}(S_i, M_{i-1})$ captures the maneuver execution sequence M_{i-1} using LSTM layer.

6.2 Results of learning from previous maneuvers

The average F1 score and the accuracy of all four models on the validation sets are show in Table 1. Both variants that extract information from M_{i-1} show significant improvements. The overall best score is obtained by G_{lstm}, which uses an LSTM layer for capturing M_{i-1}. Various network configuration are tested with different regularization parameters applied to each input path individually. In all settings, using extra information from M_i always leads to improvement of the prediction performance.

The improvement observed for $G_s(S_i, M_{i-1})$ confirms that the statistical values of a maneuver execution contain information about driving style. This is also the traditional approach for characterizing a driver. The additional improvement obtained by $G_{lstm}(S_i, M_{i-1})$ indicates that there is more information about the driver, which can be extracted by considering the raw maneuver execution instead of only using its statistical values.

6.2.1 Overfitting

Reviewing the cross-validation results, we observe the improvement in terms of F1 score and accuracy of $G_s(S_i, M_{i-1})$ and $G_{lstm}(S_i, M_{i-1})$ over $G(S_i)$ in eight out of ten folds. The maximal improvement reaches 18.5 % in F1 score which translates to 9.4 % accuracy. Fig 5a shows the validation score of all three models in one of these eight folds. Here we can see that $G(S_i)$ gets stuck and its best score is quite low, whereas both, the G_s

Table 2: Comparison on the impact of extracting information from different maneuvers

Used Maneuver	**F1 Score**	**Accuracy**
Lefturn 1	90.2 %	93.7 %
Lefturn 2	90.0 %	93.6 %
Roundabout	91.1 %	94.7 %

and G_{lstm} models benefit from the extra input M_{i-1} and reach much higher accuracy. In Fig 5b we observe the first slight effect of overfitting as the score of G_{lstm} and G_s keeps fluctuating and overfit on training data after 40 epochs. To deal with such effect, early stopping was also used for training the final model.

6.2.2 Evaluation using different past maneuvers

As mentioned in Section 5, our test route also consists of other maneuvers that can be used as input for the model. In this section, we evaluate and compare the impact of using two other left turn maneuvers as M_{i-1}in predicting taken gaps. The results are produced using the LSTM architecture in Section 2 — $G_{lstm}(S_i, M_{i-1})$. Table 2 shows the F1 and Accuracy scores on predicting the taken gap at main left turn scenario. Overall, the system still benefits the most when it extracts information from roundabout maneuvers. Since we are predicting taken gaps at a left turn maneuver, the first intuition would be the system should gain information of we use other past left turn as input. However, we have to note that these two left turn maneuvers are located in a 30-zone with low traffic, whereas the main left turn is located at 50-zone with high-traffic roads. The behavior of the these two left turns are thus different from the main left turn. On the other hand, the roundabout maneuver is a complex one which is longer and require more inputs from driver, therefore there should be more information that can be extract from a roundabout maneuver that is helpful for predicting taken gap

7 Conclusion

In this work we proposed a new approach to learn the dependency between maneuver execution, namely to extract the information about driver behavior and style, and use it to improve the performance of the prediction task in driver assistance systems. We implement our approach using neural networks as building blocks and empirically evaluate the model in a left-turn scenario using on-road data. The results show that the model is able to extract the driver impact from the past maneuver executions and can use it to improve the prediction by more than 9 % in terms of F1 score.

In general our approach can be used to evaluate the amount of information from driver that can be extracted from a maneuver execution. We compared the effect of using different types of past maneuver on predicting taken gap. The results show that roundabout maneuvers contain more individual information from driver. From that the gap prediction task benefit more than from using turning maneuvers in 30-zone.

References

[1] C. C. Aggarwal, *Recommender Systems: The Textbook*. Springer-Verlag, 2016.

[2] A. Rosenfeld, Z. Bareket, C. V. Goldman, D. J. Leblanc, and O. Tsimhoni, "Learning drivers behavior to improve adaptive cruise control," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 19, no. 1, pp. 18–31, jan 2015.

[3] P. S. Fancher and J. R. Sayer, "A Comparison of Manual Versus Automatic Control of Headway as a Function of Driver Characteristics," in *Proceedings of the 3rd Annual World Congress on Intelligent Transport Systems*, 1996.

[4] Z. Constantinescu, C. Marinoiu, and M. Vladoiu, "Driving Style Analysis Using Data Mining Techniques," *International Journal of Computers Communications & Control*, vol. 5, no. 5, p. 654, dec 2010.

[5] M. Ishibashi, M. Okuwa, S. Doi, and M. Akamatsu, "Indices for characterizing driving style and their relevance to car following behavior," in *Proceedings of the SICE Annual Conference*. IEEE, sep 2007, pp. 1132–1137.

[6] D. J. French, R. J. West, J. Elander, and J. M. Wilding, "Decision-making style, driving style, and self-reported involvement in road traffic accidents," *Ergonomics*, vol. 36, no. 6, pp. 627–644, jun 1993.

[7] V. Butakov and P. Ioannou, "Personalized Driver/Vehicle Lane Change Models for ADAS," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4422–4431, oct 2015.

[8] P. J. Cooper and Y. Zheng, "Turning gap acceptance decision-making: the impact of driver distraction," *Journal of Safety Research*, vol. 33, no. 3, pp. 321–335, oct 2002.

[9] C. M. Martinez, M. Heucke, F. Y. Wang, B. Gao, and D. Cao, "Driving Style Recognition for Intelligent Vehicle Control and Advanced Driver Assistance: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 666–676, mar 2018.

[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, nov 1997.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[12] H. Q. Dang, J. Fürnkranz, A. Biedermann, and M. Hoepfl, "Time-to-lane-change prediction with deep learning," in *Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems (ITSC-17)*. IEEE, oct 2017, pp. 1–7.