2019 IEEE Intelligent Vehicles Symposium (IV)
Paris, France. June 9-12, 2019

Modeling dangerous driving events based on in-vehicle data using Random Forest and Recurrent Neural Network

Daniel Alvarez-Coello[*,†], Benjamin Klotz[*,‡], Daniel Wilms[*],
Sofien Fejji[*], Jorge Marx Gómez[†], and Raphaël Troncy[‡]

Abstract— **Modern vehicles produce big data with a wide variety of formats due to missing open standards. Thus, abstractions of such data in the form of descriptive labels are desired to facilitate the development of applications in the automotive domain. We propose an approach to reduce vehicle sensor data into semantic outcomes of dangerous driving events based on aggressive maneuvers. The supervised time-series classification is implemented with Random Forest and Recurrent Neural Network separately. Our approach works with signals of a real vehicle obtained through a back-end solution, with the challenge of low and variable sampling rates. We introduce the idea of having a dangerous driving classifier as the first discriminant of relevant instances for further enrichment (e.g., type of maneuver). Additionally, we suggest a method to increment the number of driving samples for training machine learning models by weighting the window instances based on the portion of the labeled event they include. We show that a dangerous driving classifier can be used as a first discriminant to enable data integration and that transitions in driving events are relevant to consider when the dataset is limited, and sensor data has a low and unreliable frequency.**

I. INTRODUCTION

Smart devices rely on high-quality data sources which can be located and accessed remotely. Thanks to the increasing number of connected devices, new applications can combine multiple domains. The automotive industry also follows this trend with its connected vehicles [1]. Nevertheless, there is still the need for international open standards and protocols to enable uniform data interaction [2]. Initiatives, such as the Data-Centric Manifesto[1], show the interest in data-driven solutions regardless of the domain. Apart from the fact that it would be too costly to share all raw vehicle sensor data to the cloud, an application developer would have to adapt their application throughout the variety of many models and brands. Also, vehicles produce big time-series data [3] which increases the complexity of data processing pipelines handled differently by each application.

One could extract and communicate the meaning of vehicle data instead of sensor values to eliminate the need for in-depth domain knowledge for making new applications. This abstraction process corresponds to the transformation of data into information and is part of a Data, Information, Knowledge, Wisdom (DIKW) hierarchy which describes the building blocks for reasoning [4]. Our goal is to simplify vehicle data into information that describes how the driver and vehicle behave. Thus, we focus on modeling dangerous driving events using machine learning to classify past time windows. In this work, we assume that just dangerous driving events are relevant to consider for further enrichment.

This paper is organized as follows: we discuss current approaches and models related to the generation of information on driver behavior in section II. Our approach is discussed in section III, followed by the implementation details in IV. The evaluation results of the classifiers and our experiments to test them are presented in section V. We conclude with the principal findings and possible future directions in section VI.

II. RELATED WORK

There are different approaches to Driver Behavior Modeling (DBM). Some authors study driving patterns based on the driver's comfort [5], or physiological signals [6]; whereas the majority focus on cameras, in-vehicle signals, and smartphones [7]. We consider only existing vehicle signals that could be adapted in the future to possible standards in development such as VSS[2] or ontological models like VSSo[3] or the driving context ontology [9].

Most of the related work focuses on individual actions of driving. The combination of such actions defines relevant behavioral domains such as drowsiness [10]–[12], distraction [13], [14], and aggressiveness [15]–[20] (see figure 1).

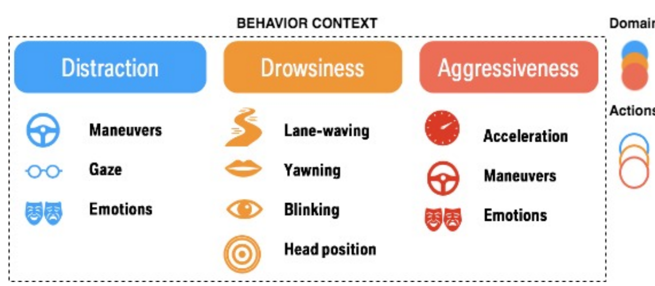


Fig. 1. Main domains and actions in Driver Behavior Modeling

A. Using Cameras

In recent years, computer vision applications using deep neural networks have shown important advances in image

[*] BMW Research, New Technologies, Innovation. Garching, Germany.
[†] Department of Computer Science. University of Oldenburg. Oldenburg, Germany. daniel.alvarez@uni-oldenburg.de
[‡] Department of Data Science. EURECOM. Sophia Antipolis, France.

[1] http://datacentricmanifesto.org

[2] Vehicle Signal Specification https://w3.org/auto/wg/wiki/Vehicle_Signal_Specification_(VSS)/Vehicle_Data_Spec

[3] Vehicle Signal Ontology [8]

and video processing. In the driving context, applications can recognize different driver actions (e.g., driver's gaze and head position[13], [14], eye blinking[21], yawning[10], emotions [22], etc.) and surrounded elements (e.g., traffic signs, pedestrians, road lanes, other vehicles, etc. [23], [24]). Although they are not under the scope of our study, such information could be used for further data enrichment.

B. Using Vehicle Signals

The access to vehicle signals is usually restricted and requires specific setups and dedicated hardware such as an On-Board Diagnostics (OBD) device. Depending on the signals of interest, an alternative is to use a vehicle simulator. Some authors use simulated data to detect aggressive driving events. [15] applies SVMs and K-means clustering. Similarly, [16] includes more signals and uses a semi-supervised learning approach.

There are also efforts to characterize the driver's profile. [25] uses signals to identify the driving style after recognizing the type of maneuver. For this purpose, logical conditions are applied to the sensor values to determine the status of the vehicle. However, their dataset is not available. Martinez et al. [26] present an approach for identifying the driver, too. While it provides an excellent foundation to learn how to differentiate the drivers, it does not specify the behavioral patterns. Burton et al. use the Euclidean distance traveled and the average speed of the vehicle to discriminate driving styles. Driver profiling does not have the granularity we look for, because it focuses on the behavior over time and not on single events.

C. Using External Sensors

Since accessing vehicle data can be challenging, some researchers opt for low-cost external devices as an alternative (e.g., smartphones, Raspberry Pi, etc.). Such devices are equipped with inertial sensors like accelerometers, gyroscope, magnetometer, camera, and others.

[11] explains the implementation of a mobile application that detects drowsiness and aggressiveness. It rates the driver's behavior and provides life feedback about the driving patterns. Drowsiness is given by lane drifting and weaving events using computer vision, where the tracking of lane marks of the road determine how centered the vehicle is. On the other hand, aggressiveness is inferred purely by the accelerometers. The number of critical events defines the level of distraction that influences the driver. The system works only for speeds higher than 50 km/h.

[17], [18] use a smartphone as the sensing and processing device to classify aggressive events. They use an end-point detection algorithm, as well as Dynamic Time Warping which is computationally expensive. A drawback is that the entire event needs to happen before the system can process it. On the other hand, [27] classifies the trajectory with the aid of a highly accurate GPS data logger as either smooth or aggressive. It uses a mathematical model, and its solution is not suitable for a real-time application.

Another approach [28] aims to detect dangerous driving based on four actions: abnormal speeding, steering, weaving, and using the phone while driving. Nevertheless, no data is collected, and the decisions rely on experimentally predefined thresholds. Similarly, [29] uses also thresholds together with an end-point algorithm to detect driving events, obtain statistical features, and classify them using a neural network.

One way to achieve the granularity we wanted is by classifying the maneuvers that the driver performs. The work by Junior et al. [19] explores this topic as well as the application of machine learning techniques to classify driving events based on aggressiveness. Their dataset is publicly available which facilitates the analysis. They use a smartphone to collect 3-axis signals from accelerometer, gyroscope, and magnetometer. This approach was used as the primary reference for our study. With the dataset of [19], Carvalho et al. [20] investigate the use of Recurrent Neural Network (RNN) to classify maneuvers.

III. APPROACH AND CLASSIFICATION ASPECTS

We considered the work from Junior et al. [19] as the basis for our approach. After replication of their grid-searches using their public Driving Behavior Dataset[4], we corroborate their conclusions that Random Forest (RF) outperforms Support Vector Machine, Neural Network, and Multi-Layer Perceptron in the multi-class maneuver classification task. Therefore, we select RF as the first technique to test. Since we are dealing with sequences, RNN are also considered [20].

A. Base Classifier

In contrast to [19], we propose to split the multi-class classification problem into two parts as shown in figure 2. A binary classifier of dangerous driving will tell us what time-window instances are relevant for further processing. Then other criteria (e.g., type of maneuver) can be used to enrich the outcomes of the base classifier. In this way, results of different applications could also be added (e.g., driver's emotion of the last seconds, drowsiness percentage, gaze, etc.).

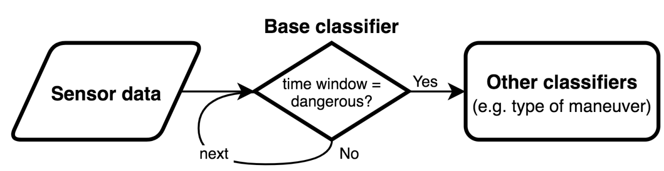


Fig. 2. A base classifier detects a dangerous situation by using relevant vehicle signal data. More specific classifiers can enrich the outcome

B. Feature Selection and Extraction

Based on [26], [30], we selected the subset of 12 signals shown in the table I. We have two types of variables: continuous and categorical. For continuous signals, we extract statistical features (i.e., mean, median, standard deviation,

[4]https://github.com/jair-jr/driverBehaviorDataset

and trend [19]). For categorical signals we take only the median value. All 12 signals were used in RF. For RNN, we did not use 3 of the signals because they were ranked as less important in the analysis with Random Forest: displayed speed, gear, and brake DSC safe.

Continuous signals	Categorical signals
Lateral acceleration	Acceleration efficiency
Longitudinal acceleration	Gear *
Accelerator pedal position	Brake pressed
Actual speed	Brake Dynamic Stability Control (DSC) state *
Speed displayed *	
Engine consumption	
Engine RPM speed	
Engine torque	

TABLE I

SELECTED SIGNALS. THOSE MARKED WITH "*" WERE NOT CONSIDERED FOR RNN

C. Instance Relevance

One driving event $\boldsymbol{E}$, in our case a maneuver, is composed of a sequence of measurements with a duration of $\boldsymbol{e_{size}}$. To classify the events, the size of the time window $\boldsymbol{w_{size}}$ should generalize for all the maneuvers of interest. The ones we want to classify are in the range of a few seconds. Thus, we tried out window sizes between 1 and 10 seconds.

Nevertheless, the low and irregular sampling frequency we get at the back-end added complexity to the classification of short driving events. Especially because sometimes time windows do not contain enough values to constitute a sample suitable for training. To overcome the limitation, we propose to consider the transitions between driving events as valid instances for training.

When $\boldsymbol{W}$ hops over time, it will not always be covering the whole driving event (i.e., $\boldsymbol{W}$ partially overlapping $\boldsymbol{E}$). A window $\boldsymbol{instance}$ is when $\boldsymbol{W}$ is at a specific time step. We only care about instances when the window is overlapping the occurrence of $\boldsymbol{E}$. With that said, the total number of instances of one driving event $\boldsymbol{instances_{total}}$, and the instance index $\boldsymbol{i}$ are given by:

$$instances_{total} = w_{size} + e_{size} - 1 \tag{1}$$

$$i \in \{1, 2, 3, ..., (w_{size} + e_{size} - 1)\} \tag{2}$$

Since one maneuver has several instances, such instance have different relevance. To determine the importance of the instances, we introduce a method to calculate their relevance as a number between 0 (not important) and 1 (most relevant). It considers the following aspects:

- When $w_{size} = e_{size}$, there exists just one instance with a relevance of 1.
- When $w_{size} > e_{size}$, relevance is 1 for all instances in which all the frames of $\boldsymbol{E}$ are covered by $\boldsymbol{W}$.
- When $w_{size} < e_{size}$, relevance is 1 for instances in which $\boldsymbol{W}$ is inside $\boldsymbol{E}$.

$$relevance = \frac{n}{w_{size}} \cdot \frac{n}{e_{size}} \cdot k = \frac{n^2}{w_{size} \cdot e_{size}} \cdot k \tag{3}$$

$$n = \begin{cases} i+1, & \text{if } (i < w_{size}) \text{ and } (i < e_{size}) \\ i-1, & \text{if } (i > w_{size}) \text{ and } (i > e_{size}) \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

$$k = \begin{cases} \frac{w_{size}}{e_{size}}, & \text{if } (w_{size} > e_{size}) \\ \frac{e_{size}}{w_{size}}, & \text{if } (w_{size} < e_{size}) \\ 1, & \text{if } (w_{size} = e_{size}) \end{cases} \tag{5}$$

D. Random Forest Parameters

In addition to [19], we add the window size and the instance relevance as our custom parameters. We do a grid-search for RF to find the best parameters from the table II.

Custom parameters	
Window size [frames]	{2, 3, 4, ..., 10}
Minimum instance relevance	{0.1, 0.2, ..., 1.0}
Random Forest	
Number of estimators	{10, 11, 12, ..., 25}
Maximum features	{10, 15, "log2"}
Maximum depth	{5, 10, 15}

TABLE II

PARAMETERS TESTED IN THE GRID-SEARCH FOR RF

E. Recurrent Neural Network Parameters

For RNN, we trained different combinations of parameters based on [20] (see table III). We used a window size of 10 frames and an instance relevance of 0.7. The number of epochs was 500 with an early stopping patience of 50 epochs. The optimizer was "RMSprop" and the learning rate 0.001.

Recurrent Neural Network	
Number of hidden layers	{1, 2}
Number of recurrent units in the hidden layer	{10, 15, 16, 32, 64, 128}
Recurrent unit type	{LSTM, GRU}
Dropout	{0.1, 0.2}
Recurrent dropout	{0.1, 0.2}

TABLE III

PARAMETERS TESTED ON THE RNN IMPLEMENTATION

IV. IMPLEMENTATION

A. Dataset

We collected and labeled vehicle data of two licensed drivers. The maneuvers were the same as in [19] (i.e., aggressive and normal turns, lane changes, accelerations, and braking). The signals come from the CAN which is accessed via a dedicated back-end architecture that is developed within BMW Research. The collection of data is only used in research for tests like the one conducted in this study.

Once the data was collected, we downsampled the series to half-second periods by assigning the aggregated values to the starting point of the current time window. This frequency was determined based on the lowest rate among the selected signals.

B. Considerations for Training

- We use the Area Under the ROC curve (AUC) [31] as the evaluation metric for the trained models since it is a trade-off between False Positive Rate and True Positive Rate by considering all the possible thresholds for the classification. AUC is a better metric for classification problems that have an imbalanced number of samples.
- We joined left and right lane-changes to deal with low sampling frequency issues because those maneuvers are the shortest in duration. Lateral acceleration was inverted to double the number of samples in this class.
- We used binary cross-entropy as the loss function for driving mode classification and categorical cross-entropy for the maneuver classification.
- To use RNNs, we first normalized our data according to the minimum and maximum possible values of the signals. The input layer is feed with sequences of 10 measurements from 9 signals.

V. RESULTS

In this section, we present the results of the best classifiers found for our specific use-case and the corresponding experiments that were conducted.

A. Classifiers Evaluation

The grid-search on RF showed us that models which use instance relevance were ranked among the best combinations. The parameters corresponding to the best RF found are presented in table IV.

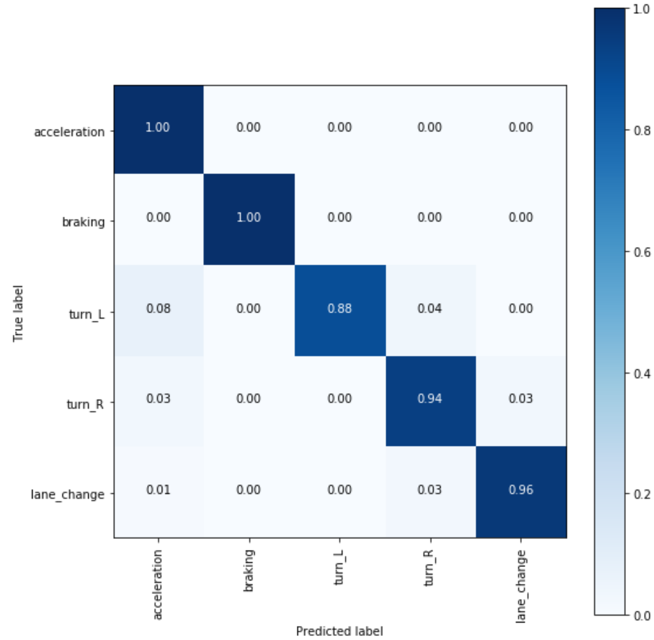
Parameter \Classifier	Base	Maneuver
Window size [frames]	10	10
Minimum instance relevance	0.9	0.8
Number of estimators	15	24
Maximum features	5	“log2”
Maximum depth	10	15

TABLE IV

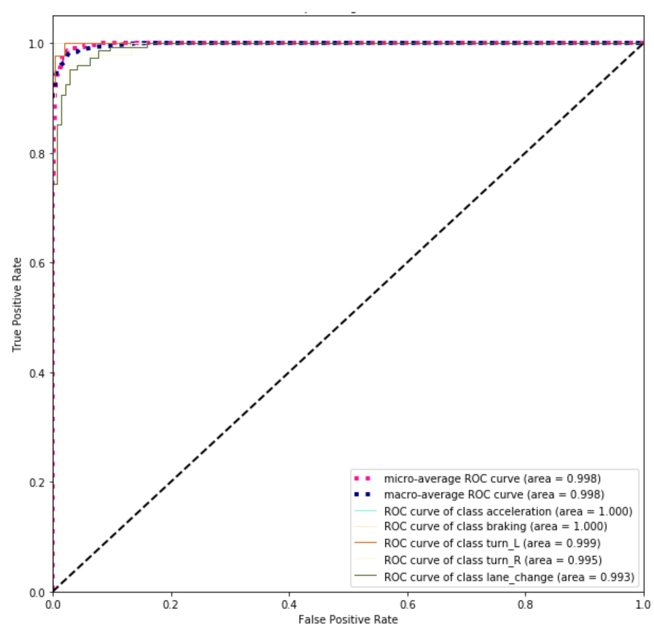
PARAMETERS OF THE BEST RF FOUND

Regarding RNN, one hidden layer with 64 recurrent units showed a better score for both classification tasks. The output layer contains 2 and 5 units for driving mode and maneuvers respectively. LSTM cells did slightly better than GRU for most of the tested combinations.

For the base classifier, both RF and RNN classified the instances of the test set correctly. While for the maneuver classification, the corresponding confusion matrices and the ROC curves show a few miss classifications of turns and lane changes (see figure 3). Nevertheless, the consideration of lane-changes to both sides as just one class (doubled samples by inverting the sign of the lateral acceleration) improved the performance significantly compared to the first attempt when we tried to classify lane change to the right and left separately.



(a) Normalized confusion matrix



(b) Multi-class ROC curve

Fig. 3. Evaluation of dangerous maneuver classifier using RNN

B. Test Experiments

We tested the best found models with 10 unseen trajectories. For this purpose, we used two different routes of a track where each trajectory corresponded to one lap. The test drivers were given specific instructions (see table V) on how to drive before each lap. The instruction of 3 driving styles refers to 3 laps on a given route, where each lap had a different style (i.e., normal, moderate, aggressive).

For every time step, the base classifier will predict the class of the previous 10 frames, the overall danger score is

Route	Driver	Instruction
1	A	3 driving styles
1	B	3 driving styles
1	B	2 laps of free driving
2	A	3 driving styles

TABLE V
TEST TRAJECTORIES AND THEIR CORRESPONDING INSTRUCTION

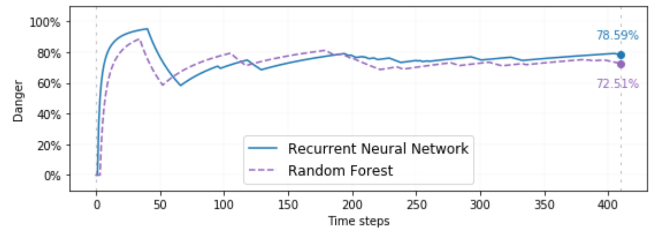
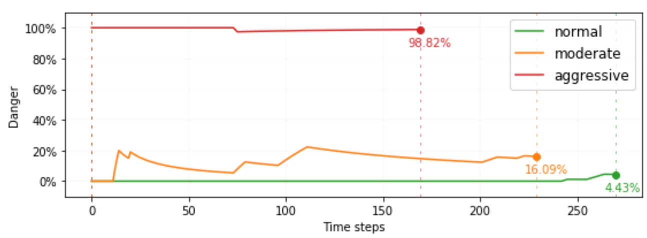
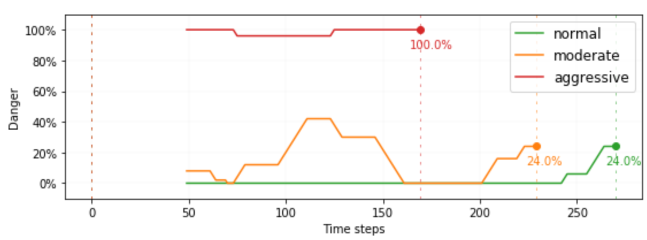


Fig. 5. Danger score of Driver B on route 1 (2 laps of free driving)

calculated by dividing the total number of positive outcomes by the total number of time steps. If we want to have more granularity, we can calculate a moving score by considering only a given amount of previous time steps. As we see in figure 4, the overall danger score of the RNN base classifier reflects the instructions given to the driver for the 3 driving styles: normal (lower-level), moderate (mid-level), and aggressive (upper-level). Likewise, the moving score provides us more insight into how the behavior over time is.



(a) Overall danger score



(b) Score of the past 50 frames

Fig. 4. Danger score of Driver A on route 1 (3 driving styles) using RNN

Additionally, we collected 2 laps of free driving and compared the model's outcome against the perception of two co-pilots who were inside the vehicle. The co-pilots wrote down their perception of danger in a scale from 0 (no danger) to 4 (most dangerous). The average danger score perceived by the co-pilots was roughly 67%, which is not far from the approximately 75% predicted by the models (see figure 5).

Since we know the sequences of the maneuvers performed on the track, one can map the outcomes of an unseen trajectory (i.e., data that is new to the trained model) to check how consistent they are. Figure 6 shows how the classified maneuvers of an aggressive lap match the sequences of the track.

VI. CONCLUSION

A binary classifier of dangerous driving events based on in-vehicle signals can simplify vehicle data and enable the integration of other domains. Compared to state-of-the-art methods, the proposed approach can provide similar results with lower and variable sampling rate. The instance relevance let us use samples in which a hopping window overlaps the driving event partially.

The collection of dangerous driving samples is time-consuming and requires special considerations (i.e., dedicated track, qualified drivers, correct labeling, etc.). A limitation of our implementation is that the labels were selected by one person, which translates into a model that represents the labeler's danger perception. Hence, the assignation of labels should be extended to more criteria to reduce potential bias. To overcome this situation we are working on an infrastructure to involve normal test drivers outside dedicated tracks. Mainly, we use the base classifier model to recognize dangerous driving events in the back-end, let the driver rate the detected situation and reinforce the model over time with less overhead.

Sending classified driving events over the network is more practical than transferring raw data. Therefore, our next steps would be to predict incoming data streams directly in the vehicle with a reinforced model and use graph data to prioritize data relationships for the integration with other domains. One approach to deal with interoperability issues across platforms could be by mapping the detected driving events to a standardized data model, such as VSS/VSSo.

REFERENCES

[1] Cisco, "Internet of things," 2016, accessed: 2019-01-25. [Online]. Available: www.bit.ly/2vpGRxp

[2] Postscapes, "Iot standards and protocols," 2018, accessed: 2018-05-03. [Online]. Available: www.postscapes.com/internet-of-things-protocols/

[3] McKinsey, "Ready for inspection – the automotive aftermarket in 2030," *McKinsey Center for Future Mobility*, June 2018. [Online]. Available: www.bit.ly/2MEhXlG

[4] J. Rowley, "The wisdom hierarchy: representations of the dikw hierarchy," *Journal of information science*, vol. 33, no. 2, pp. 163–180, 2007.

[5] X. Chang, J. Rong, C. Zhou, and H. Li, "Relationship between driver's feeling and vehicle operating characteristics on urban road," in *Intelligent Control and Automation (WCICA), 2016 12th World Congress on*. IEEE, 2016, pp. 3033–3037.

[6] N. Li, T. Misu, and A. Miranda, "Driver behavior event detection for manual annotation by clustering of the driver physiological signals," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 2583–2588.

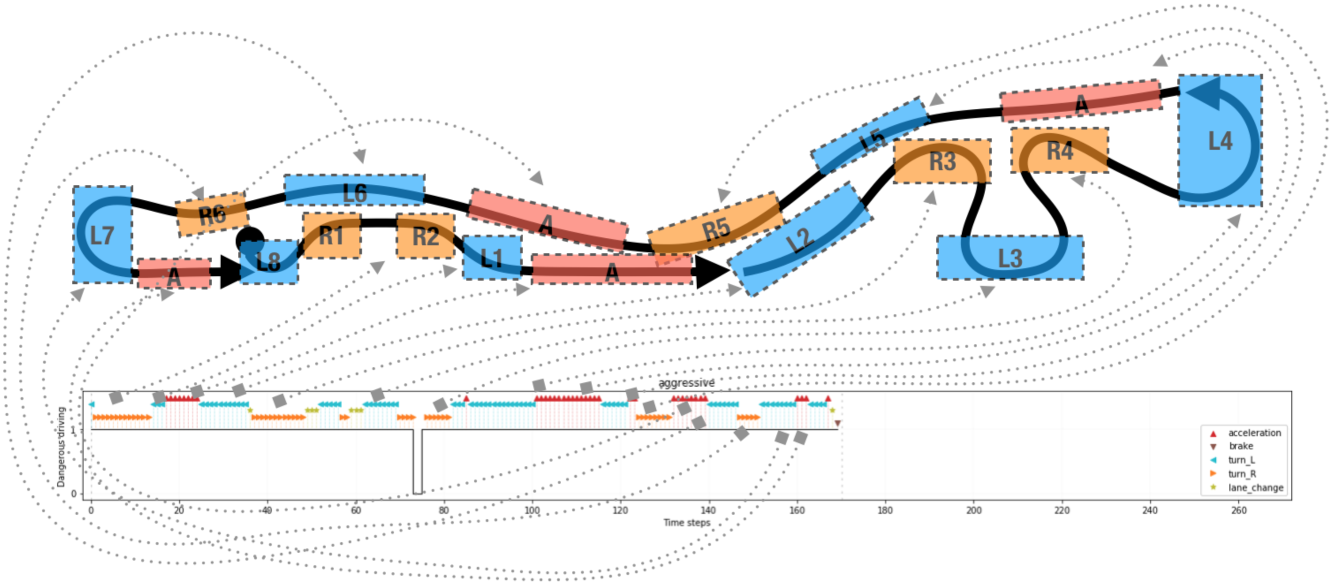


Fig. 6. Trajectory reconstruction of route 1 from the aggressive lap of Driver A using RNN

[7] N. AbuAli and H. Abou-zeid, "Driver behavior modeling: Developments and future directions," *International Journal of Vehicular Technology*, 2016.

[8] B. Klotz, R. Troncy, D. Wilms, and C. Bonnet, "VSSo - A vehicle signal and attribute ontology," in *SSN 2018, 9th International Semantic Sensor Networks Workshop, 9 October 2018, Monterey, CA, USA*, Monterey, UNITED STATES, 10 2018. [Online]. Available: http://www.eurecom.fr/publication/5691

[9] ——, "A driving context ontology for making sense of cross-domain driving data," in *BMW Summer school*, Raitenhaslach, Germany, July-August 2018.

[10] B. Reddy, Y.-H. Kim, S. Yun, C. Seo, and J. Jang, "Real-time driver drowsiness detection for embedded system using model compression of deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 121–128.

[11] L. M. Bergasa, D. Almería, J. Almazán, J. J. Yebes, and R. Arroyo, "Drivesafe: An app for alerting inattentive drivers and scoring driving behaviors," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 240–245.

[12] E. Romera, L. M. Bergasa, and R. Arroyo, "Need data for driver behaviour analysis? presenting the public uah-driveset," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 387–392.

[13] T. Liu, Y. Yang, G.-B. Huang, Y. K. Yeo, and Z. Lin, "Driver distraction detection using semi-supervised machine learning," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 1108–1120, 2016.

[14] I.-H. Choi, S. K. Hong, and Y.-G. Kim, "Real-time categorization of driver's gaze zone using the deep learning techniques," in *Big Data and Smart Computing (BigComp), 2016 International Conference on*. IEEE, 2016, pp. 143–148.

[15] W. Wang and J. Xi, "A rapid pattern-recognition method for driving styles using clustering-based support vector machines," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 5270–5275.

[16] W. Wang, J. Xi, A. Chong, and L. Li, "Driving style classification using a semisupervised support vector machine," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 5, pp. 650–660, 2017.

[17] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 1609–1615.

[18] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012, pp. 234–239.

[19] J. F. Júnior, E. Carvalho, B. V. Ferreira, C. de Souza, Y. Suhara, A. Pentland, and G. Pessin, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning," *PLoS one*, vol. 12, no. 4, p. e0174959, 2017.

[20] E. Carvalho, B. V. Ferreira, J. Ferreira, C. de Souza, H. V. Carvalho, Y. Suhara, A. S. Pentland, and G. Pessin, "Exploiting the use of recurrent neural networks for driver behavior profiling," in *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 2017, pp. 3016–3021.

[21] A. Rosebrock, "Drowsiness detection with OpenCV," 2017, accessed: 2018-01-15. [Online]. Available: www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/

[22] O. Arriaga, M. Valdenegro-Toro, and P. Plöger, "Real-time convolutional neural networks for emotion and gender classification," *arXiv preprint arXiv:1710.07557*, 2017.

[23] O. Kumtepe, G. B. Akar, and E. Yuncu, "Driver aggressiveness detection via multisensory data fusion," *EURASIP Journal on Image and Video Processing*, vol. 2016, no. 1, p. 5, 2016.

[24] S. M. Hegazy and M. N. Moustafa, "Classifying aggressive drivers for better traffic signal control."

[25] M. Van Ly, S. Martin, and M. M. Trivedi, "Driver classification and driving style recognition using inertial sensors," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 1040–1045.

[26] M. V. Martínez, I. Del Campo, J. Echanobe, and K. Basterretxea, "Driving behavior signals and machine learning: a personalized driver assistance system," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 2933–2940.

[27] A. B. R. González, M. R. Wilby, J. J. V. Díaz, and C. S. Ávila, "Modeling and detecting aggressiveness from driving signals," *IEEE Transactions on intelligent transportation systems*, vol. 15, no. 4, pp. 1419–1428, 2014.

[28] F. Li, H. Zhang, H. Che, and X. Qiu, "Dangerous driving behavior detection using smartphone sensors," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 1902–1907.

[29] P. Brombacher, J. Masino, M. Frey, and F. Gauterin, "Driving event detection and driving style classification using artificial neural networks," in *Industrial Technology (ICIT), 2017 IEEE International Conference on*. IEEE, 2017, pp. 997–1002.

[30] K. Zfnebi, N. Souissi, and K. Tikito, "Driver behavior quantitative models: Identification and classification of variables," in *Networks, Computers and Communications (ISNCC), 2017 International Symposium on*. IEEE, 2017, pp. 1–6.

[31] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

View publication stats