D^{2}CAV: A Maneuver-based Driving Dataset for Connected & Automated Vehicle Applications

Divas Grover*, Rajat Jain*, Behrad Toghi*, Yaser P. Fallah*, Gita Sukthankar†
*Networked Systems Lab., University of Central Florida, Orlando, FL
†Intelligent Agents Lab., University of Central Florida, Orlando, FL
{groverdivas, boharajat, toghi}@knights.ucf.edu, yaser.fallah@ucf.edu, gitars@eecs.ucf.edu

***Abstract*—The short-term future of the automated driving will be a hybrid scenario where both automated and human-driven vehicles co-exist in the same environment. In order to address the needs of such architecture, many technologies such as connected vehicles and predictive control for automated vehicles have been introduced in the literature. Both aforementioned solutions rely on driving data of the human driver. In this work, we investigate the currently available driving datasets and introduce a real-world maneuver-based driving dataset.**

***Index Terms*—Human driving, Driving dataset, MBC, CAV, Random Forest, SVM, Connected Vehicles, Autonomous Vehicles**

I. Introduction

Connected and automated vehicles (CAVs) have received significant attention during the last decade. Especially, the rise of artificial intelligence and sophisticated machine learning algorithms speeded up the research and development of the CAVs. Commercial level-4 autonomous vehicles [1] are expected to show up in the market in early to mid 2020s which will lead to experiencing a hybrid AI-human scenario. In such hybrid scenarios, autonomous and human-driven vehicles co-exist in the same road infrastructure and, as a matter of fact, interact with each other. The aforementioned interaction translates to the concept of situational awareness for autonomous vehicles in which their acts and decisions explicitly consider the behavior of a human-driven car as well. Furthermore, by creating an understanding of human driving patterns, autonomous vehicles are able to act in a predictive and proactive fashion in order to prevent crashes and safety-critical situations.

On the other hand, in the case of connected and cooperative vehicles, agents share their situational awareness over the ad-hoc vehicular networks (VANETs) utilizing vehicle-to-vehicle (V2V) communication. Recently, the author in [2] suggested a novel methodology for V2V communication as the model-based communication (MBC). The background idea behind MBC is to utilize an abstraction of the vehicles' situational awareness, i.e., an abstract model of their state, as an alternative for the current standard raw-data communication.

A vehicle's mobility patterns can be classified mainly into a dozen of short-term, or small-scale, maneuvers. Among which, one can refer to maneuvers such as U-turns, lane changes, left (and right) turns, hard-brakes, joining (and leaving) a platoon, take-over, etc. Precise detection of such maneuvers enables us to address the aforementioned use cases as well as other potential applications. Modeling a maneuver provides us with an abstract representation of the vehicle's state which can be utilized in the earlier discussed MBC architecture. On the other hand, in a hybrid AI-human scenario, recognizing a remote human-driven vehicle's intention to perform a maneuver will enable autonomous vehicles to predict and react accordingly [3], [4].

A wide variety of sensory data is available from a vehicle's Controller Area Network (CAN) bus which can be utilized to model the mobility patterns and driving maneuvers. As an illustration of the most common features, one can name steering angle, engine speed, GPS coordinates and heading, and throttle position. Every specific maneuver has different class-correlation with the features and can be highly correlated to one and uncorrelated from the other. As an illustration, a u-turn maneuver is more significantly observable in the steering-angle/heading space and a hard brake maneuver stands out in the ground speed space. Figure 1 shows the steering angle pattern in a given pair of u-turn and left-turn maneuvers.

Hence, it is expected that expressing driving maneuvers using the above mentioned features will include information redundancy. From the information theory point-of-view this redundancy enables us to achieve high accuracy both in

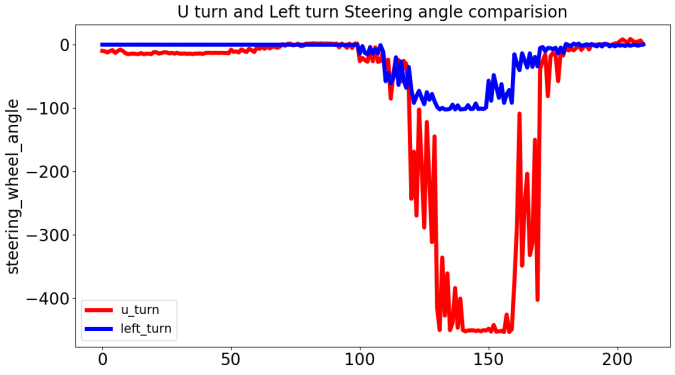

Fig. 1. A comparison between a pair of given u-turn and left-turn maneuvers and their pattern in the steering angle space

This work is funded by a research grant from Ford Motor Company, Research and Advanced Engineering (R&A) under OpenXC project.

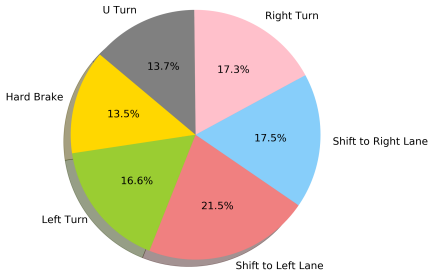

Fig. 2. Unbalanced dataset distribution for containing maneuvers

classification and regression operations. A few real-world driving datasets currently exist in the literature which contain recorded {GPS+CAN} data from human driven vehicles in urban and highway environments. Among which, we can name the NGSIM dataset [5] recorded in Ann Arbor and Greater Detriot area, 100-car near-crash dataset [6] which focuses on critical near collision scenarios and can be used for the safety related applications such as forward collision warning (FCW), and SPMD dataset [7] which is extracted from video footage of a highway and includes short-term maneuvers of vehicles in camera's field of view.

However, none of the mentioned works have parsed and labeled the data into specific maneuvers which adds a burden for the researchers to tentatively label the maneuvers in the post-processing fashion. One main downside of post-process data labeling is the low reliability and probable false labeling which can degrade the desired regression or classification application's performance. In this work, we present a maneuver-based real-world driving dataset for the CAV applications, titled *Driving Dataset for Connected and Automated Vehicles* (D^2CAV). The D^2CAV datasaet contains a large set of logged CAN bus and GPS data from human-driven vehicles performing a variety of maneuvers in the Orlando metro area, FL. We limited our interest mainly to a narrowed set of maneuvers, i.e., left (and right) turns at intersections, u-turns, hard-brakes, lane changes, and approaching intersections.

The rest of the paper is organized as follows. In Section 2, we demonstrate the field test and data collection process and describe the dataset architecture. In Section 3, we focuse on the implemented maneuver classification algorithms and present the results and analysis before concluding the paper in Section 4.

II. In-field Data Collection Campaign

For the purpose of data collection, we utilized the Ford OpenXC [8] platform and a Garmin Map-62s handheld unit as the logging tools. Three drivers with different driving styles (aggressive, mild, and conservative) are employed to drive a 2018 Ford Focus with electric assist steering and drive-by-wire throttle actuator. We conducted ~ 1000 minutes of urban and highway driving around the UCF campus in the metro Orlando area. During the field test, a co-pilot was assigned to manually label the maneuvers using a custom-made logging interface.

The logged data fields include engine speed, total fuel consumption since restart, odometer, accelerator pedal position, torque at transmission, steering wheel angle, vehicle speed, and fuel level recorded by the OpenXC logger and latitude, longitude, ground speed, and heading recorded by the Garmin handheld. As a matter of fact, this large number of features provide us with a set of redundant data which can potentially improve the performance of the application implemented on top of our dataset.

Our setup includes the OpenXC logger connected to the vehicle's OBD II connector, the handheld Garmin GPS mounted on the windshield, and the labeling operator. Prior to performing a driving maneuver, the driver notifies the co-pilot about his intention asking him to log the label via the logging interface. The logging interface automatically acquires the timestamp and records the label to be used in the post-processing stage. Different maneuvers can take different lengths of time, as an example, a u-turn is usually a longer maneuver (in time) comparing to a hard-brake. Hence, we set a $\pm 10s$ window for each maneuver and parse the trip data into smaller sub-trips, which each contain an isolated driving maneuver. The data is organized in a straight-forward arrangement, each sub-folder contains the ".csv" file of the joint GPS+CAN data of the maneuver as well as the time-domain plots of the features and a schematic of the sub-trip, i.e., the geographic representation of the maneuver.

Figure 3 shows the recorded driving path for the full dataset illustrated on top of the Google Earth™, which obviously contains a wide variety of driving conditions and maneuvers. Considering the fact that we utilized two different data sources, i.e., Vehicle CAN bus and GPS device, leads us to a challenge in the data collection campaign. The two above-mentioned data sources not only have different data rate but also are not synchronized in the time-domain. In fact, the GPS logs have an average update rate in the order of 1 Hz, where the CAN bus data is mostly consistent on 10 Hz. Thus, aggregating the CAN and GPS data logs is not straight-forward and in order to address this issue, we interpolated (up-sampled) the GPS logs and synthetically created timestamps to match with

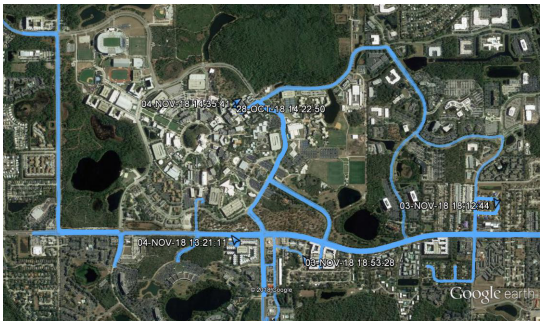

Fig. 3. A sample view of the driving path during the data collection campaign in the UCF campus (map courtesy of Google Earth™)

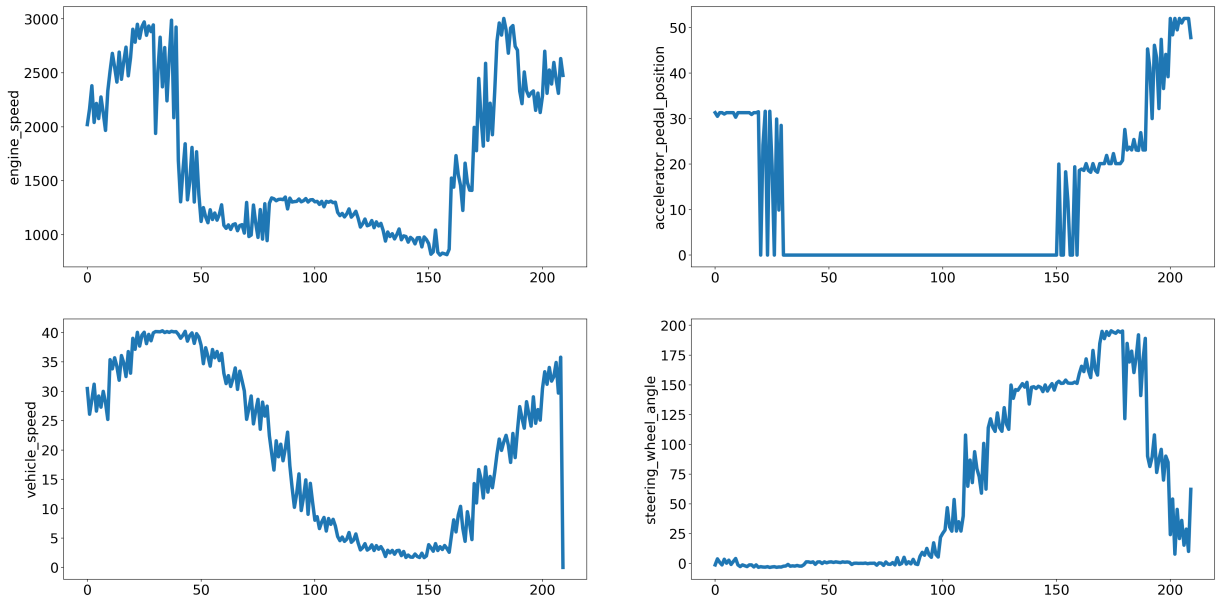

Fig. 4. Illustration of a given right-turn maneuver in terms of 4 features, represented in time-domain.

the CAN logs. Trying different interpolation methods showed us the cubic interpolation provides us with a more realistic vehicle mobility.

To summarize the dataset introduction, we have demonstrated the unbalanced class distribution in the Figure 2. It should be noted that a higher precision is expected in applying either regression or classification methods on some maneuvers such as u-turns in comparison to the less visible (in the recorded data) maneuvers such as lane changes. This matter will be more elaborated in the next section. Every sample scenario contains the time series of the aforementioned logged features, e.g., latitude, longitude, steering angle, etc. Figure 4 shows an example sub-trip data of an arbitrary right-turn maneuver.

A. OpenXC Platform

The Ford Motor Company developed a new logging interface compatible with all new-model Ford vehicles in order to support the research requirements in the academia and industry. The project is named as OpenXC [**?**] platform and includes an OBD II CAN bus logger and data analysis API. The API can be run on both Ubuntu machines and Android cell phones which provided us with more flexibility in the data collection process.

III. Classification Methodology

As it is mentioned earlier in the text, the dataset can be utilized for both prediction (regression) and classification purposes. However, in this work we focus on the later and apply two common classification methods on the dataset and measure their performance for different maneuvers. We choose Random Forest and Support Vector Machine classifiers as the candidates and compare their performance to make a decision as the the final classifier to be used as the decision block. Figure 5 shows an overview of the system architecture of our approach.

A. Random Forest Classifier

We choose the Random Forest Classifier (RFC) [9], [10] as one of the candidates for classification as it is robust to noise and over-fitting, and works well when the features are correlated. Consider the input data represented as $\mathbf{x} = [x_1, x_2 \ldots, x_n]^T \in \mathbb{R}^{n \times d}$ which consists of features from n data points each having a dimension d. A decision tree classifier routes the input feature $x_i \in \mathbf{x}$ from the root of the tree to its leaf. The final class prediction pertaining to the feature x_i can be obtained at the leaf $L(T_j(x_i))$, where T_j corresponds to a tree with an index j. Because RFC belongs to the group of algorithms that lie within *supervised learning* algorithms, therefore, it works in 2 phases. i.e., training (offline) and testing (online). Both these phases are explicitly explained in the following subsections.

1) Training: Let $\mathbf{y}$ represent the set of labels such that $L(T) \in \mathbf{y}$. The other nodes D of the tree are characterized by a binary decision function $\phi(x)$, which can route the feature towards right or left of the decision tree. For instance, if $\phi(x) = 1$, then left sub-tree t_l is selected, whereas if $\phi(x) = 0$, then the data is routed towards the right sub-tree t_r. For a tree T, the prediction function can be recursively written as:

$$f(x|D(t_r, t_l, \phi)) = \begin{cases} f(x|t_r) & if\ \phi(x) = 0 \\ f(x|t_l) & if\ \phi(x) = 1 \end{cases} \quad (1)$$

The binary decision at the nodes can be selected either randomly or by a well-defined criterion. This criterion can be modeled by a split which can optimally separate the training

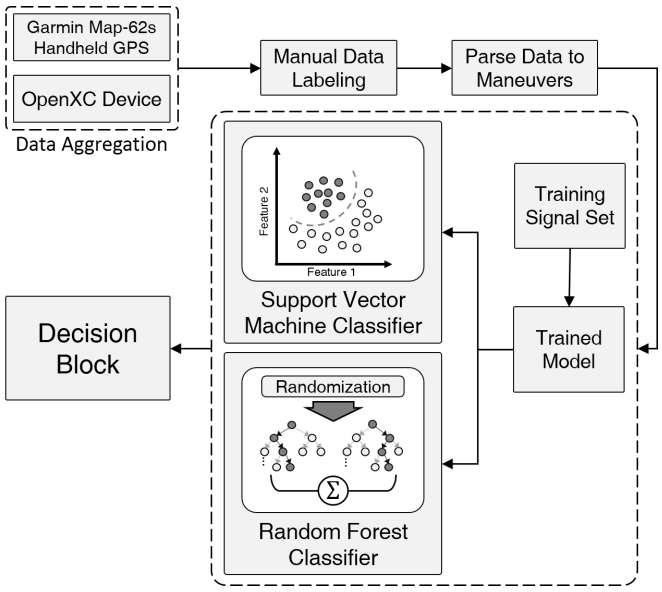

Fig. 5. Maneuver classification system architecture: Random Forest classifier and Support Vector Machine

data. This optimality can be measured by the information gain which is represented as:

$$\Delta E = -\sum_{i} \frac{|\mathbf{x}'|}{|\mathbf{x}|} E(\mathbf{x}'), \tag{2}$$

where $\mathbf{x}'$ is the partition from the data $\mathbf{x}$ and $|\cdot|$ is the size of the set. Moreover, $E(x')$ is the entropy which is represented as: $E(x') = -\sum_{j=k}^{C} p_k \ \log_2(p_k)$. Here, p_k is the proportion of examples belonging to class $k \in C$, with C denoting the number of classes.

2) Testing: While testing, the final class label y^* corresponding to a testing example $x*$ is given by:

$$y^* = \arg\max_{k \in C} \sum_{j=1}^{U} G(\, L(T_j) = k\,). \tag{3}$$

In the above equation U is the number of trees, whereas, $G(\cdot)$ is the indicator function, which is equal to 1 when $L(T(j)) = k$ and 0 otherwise.

B. Support Vector Machine

Support Vector Machines (SVMs) is one of the simple, yet effective, classifiers that can be applied on both linearly and non-linearly separable data. The SVM classifier enjoys a bound on the test error rate and can also employ complex non-linear kernels such as as Radio Basis Functions (RBF) and exponential kernels. Thus, we choose SVM as our second classifier candidate to be implemented on top of the D^{2}CAV.

The SVM classifier simply relies on maximizing the margin between the classifier hyper-plane and the support vectors. The well-known kernel trick can be utilized in order to apply non-linear hyper-planes. The SVM algorithm can be mathematically formulated as follows

$$\begin{aligned} \text{minimize:} \quad & \Phi(w) = \frac{1}{2} w^T w \\ \text{Subject to:} \quad & y_i(wx_i + b) \geq 1 \end{aligned} \tag{4}$$

where w is the weight vector, x_i is the input data and y_i is the corresponding label.

IV. Analysis & Results

As mentioned before, we implemented to classification algorithms on our dataset and carefully measured the performance of each method. Before exploring the results and analysis, it is worth mentioning that one may arise the question that why left and right turn maneuvers are being considered as two separate an d independent maneuvers while take can just simply referred to as turns. In order to address this question, we made observations on the patterns of a given left and right turn scenarios from a driving trip. The steering-angle time-series are illustrated in Figure 6 for the sake of comparison and as it is obvious from the figure, left and right turn maneuvers are not exactly symmetric. The simple reason behind this is the geometry of our roads, as an example in an intersection of a left-hand-drive road, the driver needs to traverse a larger radius circle in left-turn in comparison to the right-turn. This is also in consistency with the results shown in 6.

In order to evaluate the classification performance, we utilized three key performance indicators: F1-score, precision, and recall. Moreover, we plotted the confusion matrices to demonstrate the classification performance and errors for each of the maneuvers. Figure 9 compares performance metrics for both SVM and RF classifiers. As it is shown in the bar plots, The SVM classifier shows a noticeably lower performance, almost in all metrics, compared to the RFC case. Specifically, the SVM classifier suffers in the case of left (and right) lane changes and is not able to correctly classify most of the maneuvers. This may happen due to the intrinsic similarity between two maneuvers. On the other hand, the same problem is visible in the confusion matrix shown in the Figure 7 where the most of the misclassified data samples belong to the left and right switches. Another interesting observation can be made from the confusion matrices i.e. both SVM and RFC can classify all hard brake instances there is a very plausible

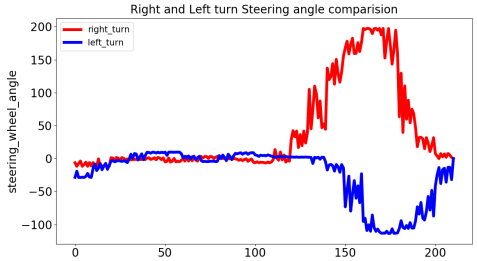

Fig. 6. A comparison between the left-turn and right-turn maneuvers and their time-domain representation.

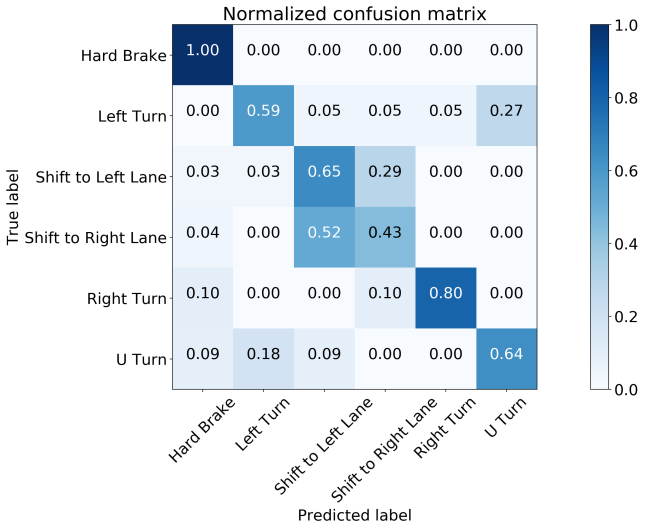

Fig. 7. The confusion matrix demonstrating the classification performance of the SVM classifier.

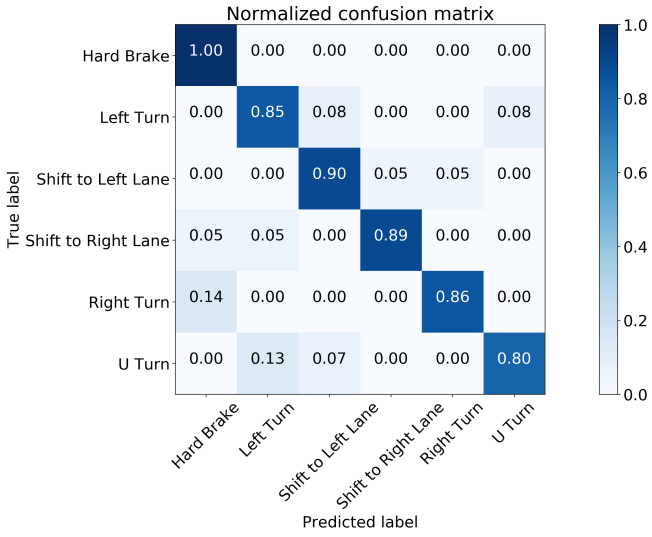

Fig. 8. The confusion matrix demonstrating the classification performance of the Random Forest classifier.

reasoning behind it, during hard brake the driver takes of their foot from accelerator pedal and puts high pressure on the brake pedal which in turn leads to accelerator pedal position being dropped to zero along side the engine speed and vehicle speed decreasing rapidly. This result from the confusion matrices is critical to safety applications in CAVs as hard brake is one critical maneuver.

However, the RF classifier performs better and demonstrates satisfactory results in terms of the classification performance. The F1-score, recall, and precision metrics are shown in the Figure 9 which shows RFC is able to maintain $> 80\%$ measure in all performance metrics. Moreover, the confusion matrix for the RFC case is plotted in Figure 8 and obviously shows a more diagonal distribution which translates to a higher classification performance. An interesting and informative discussion could be analyzing the misclassified cases as some of the maneuvers share some sub-maneuvers. As an example, a right-turn maneuver is mostly accompanied with a braking before performing the turn. Thus, if the data is not precisely parsed, we may see more misclassified data samples in this case.

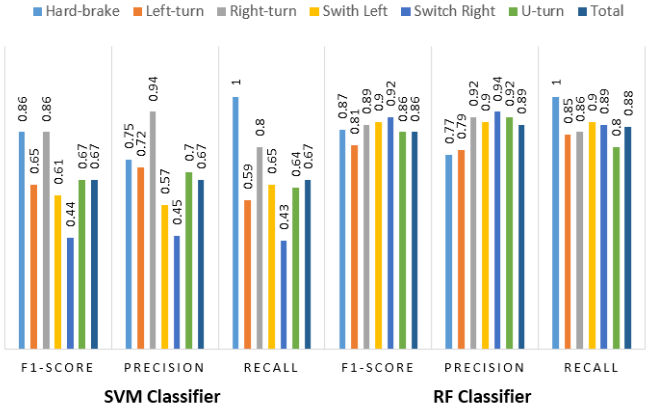

Fig. 9. Performance comparison of the RF and SVM classifiers for each of driving maneuvers.

V. Concluding Remarks

With the introduction of Connected and Automated Vehicles and consequently novel technologies for addressing the technical problems, such as the model-based communication (MBC) and predictive decision making, the need for human-driven driving datasets is arising. In this work we investigated the currently available datasets and concluded that the literature is lacking of a maneuver-based parsed dataset. We utilized the Ford OpenXC platform for our data collection campaign and recorded a preliminary set of data. Finally, two well-known classification algorithm, i.e., the Support Vector Machine (SVM) and Random Forest Classifier (RFC) are implemented on top of our dataset and their performance is evaluated on each maneuver.

VI. Acknowledgement

We would like to appreciate the support from the Ford Motor Company and OpenXC [8] research project for providing us with logging instruments and supporting this project.

References

[1] Society of Automotive Engineers. Sae (j3016) automation levels. *(SAE)*.
[2] Y. P. Fallah. A model-based communication approach for distributed and connected vehicle safety systems. pages 1–6, April 2016.
[3] H. N. Mahjoub, B. Toghi, and Y. P. Fallah. A driver behavior modeling structure based on non-parametric bayesian stochastic hybrid architecture. *IEEE Vehicular Technology Conference (VTC)*, August 2018.
[4] H. N. Mahjoub, B. Toghi, and Y. P. Fallah. A stochastic hybrid framework for driver behavior modeling based on hierarchical dirichlet process. *IEEE Connected and Automated Vehicles Symposium (CAVS)*, August 2018.
[5] Federal Highway Administration Research and Technology. Us highway 101 dataset, next generation simulation (ngsim). *https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm*.

[6] Virginia Tech Transportation Institute. 100-car naturalistic driving study. *https://www.vtti.vt.edu/PDF/100-Car*$_F act - Sheet.pdf$.
[7] Safety pilot model deployment (spmd) program.
[8] The openxc platform by ford motor company.
[9] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
[10] P. Kontschieder, S. R. Bulo, H. Bischof, and M. Pelillo. Structured class-labels in random forests for semantic image labelling. In *2011 IEEE Int. Conf. on Computer Vision (ICCV),*, pages 2190–2197, 2011.