Learning-Based Approach for Online Lane Change Intention Prediction

Puneet Kumar, Mathias Perrollaz, Stéphanie Lefèvre, and Christian Laugier

Abstract— **Predicting driver behavior is a key component for Advanced Driver Assistance Systems (ADAS). In this paper, a novel approach based on Support Vector Machine and Bayesian filtering is proposed for online lane change intention prediction. The approach uses the multiclass probabilistic outputs of the Support Vector Machine as an input to the Bayesian filter, and the output of the Bayesian filter is used for the final prediction of lane changes. A lane tracker integrated in a passenger vehicle is used for real-world data collection for the purpose of training and testing. Data from different drivers on different highways were used to evaluate the robustness of the approach. The results demonstrate that the proposed approach is able to predict driver intention to change lanes on average 1.3 seconds in advance, with a maximum prediction horizon of 3.29 seconds.**

Fig. 1.   Lane change scenario

I. Introduction

Studies of the causes of road accidents show that 57% of them are solely due to driver factors [1]. Advanced Driver Assistance Systems (ADAS) can help drivers to understand traffic situations better, take actions to make driving more comfortable, improve the traffic flow, limit the energy consumption, avoid accidents, or mitigate their consequences. For example, predicting in advance other drivers's intentions at an intersection or predicting whether a vehicle on the highway will change lanes can help a driver to understand the situation better and to avoid accidents.

This work focuses on lane change intention prediction on highways. The goal is to predict whether the ego vehicle will make a lane change and in which direction, based on sensor data. Mathematically, a vehicle trajectory can be defined by the deterministic function:

$$\Phi : \Phi(t) = \left(x(t), y(t), \frac{dx}{dt}(t), \frac{dy}{dt}(t), \frac{d^2x}{d^2t}(t), \frac{d^2y}{d^2t}(t) \right) \quad (1)$$

where the outputs of the function Φ represent the longitudinal position, lateral position, longitudinal velocity, lateral velocity, longitudinal acceleration and lateral acceleration of the vehicle respectively [2]. The trajectory segment in the interval $[T_1 \ T_2]$ is a lane change trajectory segment iff it intersects with the curve formed by the road markings. The time where the lane change occurs is defined as the time at which the trajectory segment and the lane markings intersect, as depicted in Fig. 1. In this context the problem we tackle
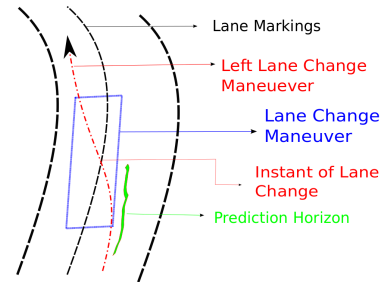
is to predict the lane change before it actually happens. In this paper a novel approach based on the combination of multiclass Support Vector Machine (SVM) and Bayesian filtering is proposed. The algorithm was tested on a real dataset collected using a passenger vehicle. The performance of the approach was evaluated by looking at the prediction horizon, the rate of false alarms, and the rate of missed detections.

The remaining of this paper is organized as follows. Section 2 reviews related work, Section 3 discusses the different aspects and constraints of the problem and then describes our approach to solve the lane change prediction problem within the discussed constraints. Section 4 presents the results. Section 5 concludes and outlines future work.

II. Related work

The term *intention prediction* can have several interpretations. Different terminologies such as *behavior prediction/recognition/identification*, *situation assessment/prediction*, *intention prediction/estimation* have been used in the literature to qualify the same type of problems. Examples of such problems are: predicting whether a driver will change lanes, predicting whether a vehicle will stop at the red traffic light etc.

In [3] the lane change prediction problem is addressed using SVMs with feature vectors consisting of the variances of the features (speed, steering angle etc.). In [4] a sparse Bayesian classifier is used for lane change intent analysis. The idea is to use time series data describing the vehicle's surrounding, the driver's head motion, and the vehicle's internal state to create a feature vector for classification. The results show that the inclusion of the driver's state (head motion) in the feature vector increases the prediction horizon. The extension of this work proposed in [5] uses Relevance Vector Machine (RVM), a Bayesian extension of SVM. It relies on information from ACC (Adaptive Cruise Control) radar, LDW (Lane Departure Warning) camera, SWA (Side

Warning Assist) radars, and head tracking camera. This approach is able to predict lane changes up to 3 seconds in advance. Another work [6] deals with intentions at road intersections and considers two types of on-road agents: compliant and violating. Two approaches were compared, the first one using an SVM-based binary classification along with Bayesian filtering and the second one using Hidden Markov Models (HMMs). The results showed that the SVM-BF based approach performed better. In [7] Coupled HMMs were trained to create models of seven different driving maneuvers. The results showed that maneuvers could be predicted on average 1 second before they actually started. In [8] a probabilistic approach using HMMs is used for situation modeling and recognition. A situation is defined as a distribution over sequences of states having a meaningful interpretation and a HMM is used to characterize each situation. The approach was tested on real data for highway driving with three situations: passing, aborted passing, and following. Similarly to this approach, in [9] a Hierarchical HMM was used for behavior recognition. The idea is to model behaviors in two layers. High level behaviors, such as: go straight, turn left, turn right, and overtake are treated as the hidden states of an HMM in the upper layer. For each high-level behavior there exists a HMM at the lower layer representing the sequence of transitions of the corresponding behavior. In [10] a Probabilistic Finite State Machine (PFSM) and fuzzy logic are used for maneuver recognition. The input variables (velocity, steering angle etc.) are fuzzified in order to estimate the basic elements (braking, halt, start etc.) constituting maneuvers. Finally a Bayesian filter is used to find the probability distribution of the basic elements of the PFSM for maneuver recognition.

III. PROPOSED APPROACH

Our approach uses lane information, speed, and steering angle for lane change intention prediction in highway scenarios. We formulate the lane change intention prediction problem as a multiclass classification problem with the three following families/classes of trajectories: *left lane change*, *right lane change*, and *no lane change*. Other situations like multiple lane changes can be seen as combinations of these three classes. A kernel-based large margin classifier known as Support Vector Machine (SVM) is used to address this problem. Major motivations behind the use of SVMs are: (a) a driver's state may lie in a high dimensional feature space [11] and a kernel maps the input data from a low dimensional space into a high dimensional space converting a nonlinear classification problem at low dimension into a linear classification problem at high dimension, (b) SVM is a maximum margin classifier, therefore it is expected to classify similar trajectories belonging to different classes reliably, (c) the objective function of an SVM is convex, therefore the solution is a global optimum, and (d) several studies (e.g. [12]) have shown that SVMs give promising results when applied to intention prediction problems for ADAS.

A Bayesian Filter (BF) is used on top of the multiclass classifier in order to improve the reliability of the predictions. It is expected that the smoothing introduced by the filter will reduce the rate of false alarms and missed detections. A generalized Bradley-Terry model [13] is used to obtain probabilistic outputs from the SVM. This probabilistic output of the SVM is fed to the BF providing a filtered and smooth output. The state transition matrix of the BF is learned from real data. The output of the BF is used for final lane change intention prediction.

Details about the proposed approach are provided in the rest of this section: (a) lane tracker, (b) multiclass probabilistic estimates using SVM, and (c) Bayesian Filter.

A. Lane tracker

To make the overall approach independent of localization devices (e.g. GPS) and digital maps (GIS), a vision and IMU based lane tracker is used to capture information about the position of the ego vehicle with respect to the road (see device in Fig. 3). In general, the lane tracking problem can be seen as a state estimation problem with the state vector consisting of the lane parameters defining the road geometry [14]. Here we use a particle filter and follow the following steps:

1) Define parametric equations describing the lane position and geometry with respect to the vehicle. The lane is assumed to have a shape similar to a clothoid curve.

2) Prediction step of the particle filter: the state vector consisting of the parameters of the parametric equations is recursively estimated using the past state vector and odometry information.

3) Image capturing and ridge detection: a ridge filter provides the low level features used in the update stage of the particle filter.

4) Update step of the particle filter: the weight of each particle is adapted to reflect how well it matches the ridge features.

B. Multiclass probabilistic estimates using SVM

Support Vector Machine (SVM) is a supervised learning algorithm that uses the concept of margin maximization for the purpose of classication. The basic idea is to find a hyperplane that maximizes the separation between the data-points of different classes. It uses the concept of kernels to project data from low dimensions to higher dimensions thus converting a nonlinear problem in low dimensions into a linear problem in higher dimensions. Given a dataset $S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)\} \in (\chi \times \gamma)^m$, with $\chi = \Re^n$ and $\gamma = \{-1, 1\}$, the convex objective function for the binary SVM is defined as [15], [16]:

$$\mathbf{w}^* = min_{\mathbf{w}, \zeta} \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{m}\sum_{i=1}^{m}\zeta_i$$

$$\text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \zeta_i, \forall i \quad (2)$$
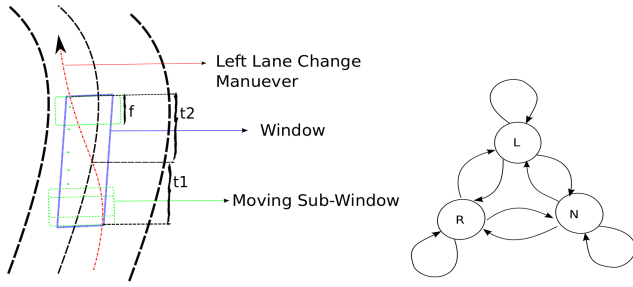
Fig. 2. Left: Sliding window approach for ground-truth feature vector generation. Right: State Space Diagram for different maneuvers

One simple approach for the multiclass extension of the above defined binary SVM is to learn k different $\mathbf{w}s$ for k classes using a one-vs-all approach.

C. Feature Selection

The discriminative capabilities of any classifier greatly depends on the selection of the feature vectors $\mathbf{x}_i$. We use feature vectors consisting of the following meta-features:

- the lateral position of the vehicle w.r.t. a lane (l)
- the steering angle of the vehicle w.r.t. the road (ϕ)
- the first derivative of l
- the first derivative of ϕ

These features were selected because their patterns are remarkably different for each of the three classes we are interested in. Since a lane change is a continuous process and can last for several seconds, the feature vector $\mathbf{x}_i$ must contain information concatenated within a time span in order to try to capture the continuity. To do so, a sliding window approach is used (see Fig. 2). A window is selected around the time when the vehicle reaches the lane markings for a lane change, all the meta-features within that window are labeled as the ground-truth data-points representing the corresponding lane change. This window size should be large enough to capture a lane change process completely. Within this window a sub-window is created and different meta-features at different times within this sub-window are concatenated to form the final feature vector of fixed length. This subwindow is moved within the window in order to generate different feature vectors representing lane changes. Mathematically, if $(t1 + t2)$ represents the window size, f represents the sub-window size, fps represents the data frames collected per second, then the feature vector at time i can be represented as:

$$\mathbf{x}_i = [\mathbf{l}_i, \dot{\mathbf{l}}_i, \Phi_i, \dot{\Phi}_i] \in \Re^n \quad (3)$$

where

$$t = \frac{1}{fps}$$
$$n = 4 \times f$$
$$\mathbf{l}_i = [l_{i-f \times t}, l_{i-(f-1) \times t}, \ldots, l_{i-t}, l_i]$$
$$\dot{\mathbf{l}}_i = [\dot{l}_{i-f \times t}, \dot{l}_{i-(f-1) \times t}, \ldots, \dot{l}_i]$$
$$\Phi_i = [\phi_{i-f \times t}, \phi_{i-(f-1) \times t}, \ldots, \phi_i]$$
$$\dot{\Phi}_i = [\dot{\phi}_{i-f \times t}, \dot{\phi}_{i-(f-1) \times t}, \ldots, \dot{\phi}_i]$$

We use a radial basis function kernel, $\mathscr{K}(\mathbf{x}_i, \mathbf{x}_j) = exp(-g\|\mathbf{x}_i - \mathbf{x}_j\|^2)$, with g as the kernel parameter and $d > 0$. The approach proposed by [13] is used to get the probabilistic outputs from the multiclass SVM. This approach extends the Bradley-Terry model for paired individual comparisons into paired team comparisons.

D. Bayesian Filter

The Bayesian filtering algorithm is the recursive form of Bayes rule which gives an inverse relationship between posterior, prior and likelihood. For lane change intention prediction, the equation to compute the probability of a maneuver M_t can be written as:

$$P(M_t|Z_{0:t}) \propto P(Z_t|M_t) \times$$
$$[\sum_{M_{t-1}=L,R,N} P(M_t|M_{t-1}) \times P(M_{t-1}|Z_{0:t-1})] \quad (4)$$

where Z_t corresponds to the observations at time t and L,R,N represent left, right and no lane changes respectively. In our approach the likelihood term in the above equation, $P(Z_t|M_t)$, is taken as the probabilistic output of the SVM. The state transition probabilities $P(M_t|M_{t-1})$ are learned offline using training data. For all the three maneuvers we get a 3×3 state transition matrix with nine unknown state transition probabilities. The corresponding state transition diagram for all the three states or classes L,R,N is shown in Fig. 2. Since in our case all the maneuvers M_t in our training data can be labeled, the state transition matrix is simply the normalization of the co-occurrences (counts):

$$S_{ML}(i,j) = \frac{S(i,j)}{\sum_k S(i,k)}$$

where, $S(i,j)$ is the number of transitions from maneuver i to maneuver j in the training data and $S_{ML}(i,j)$ is the maximum likelihood probability for $i \rightarrow j$ transition .

The posterior $P(M_t|Z_{0:t})$ is used for the final lane change intention prediction.

IV. RESULTS

A dataset with 139 lane changes was collected with two different drivers on a highway near Grenoble, France, using our Lexus LS600h experimental platform (see Fig. 3). The vehicle is equipped with a TYZX stereo camera with 22 cm baseline, $62°$ field of view, 512×320 pixels resolution and focal length of 410 pixels. The training dataset consists of 22 left lane changes, 24 right lane changes, and 24 trajectories with no lane change. The testing dataset consists of a total of 69 lane changes (left + right). The parameters of the SVM were set to $g = 0.0625$, $C = 8$, $t1 = 2$, $t2 = 2$, $f = 32$, fps = 32, and final classification rate = 5 classifications per second. The final classification rate is lower than the fps because 32 classifications per second is more than what is necessary for a real-time application. On the other hand, for training we used all the data acquired at 32 fps because data at higher fps better approximates the lane change event. The reasons for choosing the particular values for the other parameters are discussed in subsequent sections.

Fig. 3. The Lexus experimental platform used for real data collection

Fig. 4. Ground truth testing data with 5 lane changes (3 left + 2 right)

A. Evaluation metrics

The following terminologies and evaluation metrics are used:

- Ground truth: the moment at which the vehicle reaches the lane markings.
- Classes:
  Class 1 = right lane change
  Class 2 = no lane change
  Class 3 = left lane change
- Prediction point: the time at which a lane change is predicted.
- Prediction time: the time difference between the ground truth and the corresponding prediction point. In the evaluation, the prediction point nearest to the ground truth is always chosen (worst case). If the prediction point occurs later than the ground truth then this is considered as false prediction.
- Precision, recall and F1-Score: the precision is the fraction of trajectories where the classifier correctly declared class m out of all instances where the classifier declared class m. The recall is the fraction of events where the classifier correctly declared class m out of all of the cases where the true class was m. The F1-Score is the harmonic mean of the precision and the recall.

B. Classification Results

Fig. 4 shows the ground truth for 5 lane changes (3 left + 2 right) belonging to the testing data. The final predicted classes for the same data using SVM alone and SVM + BF are shown in Fig. 5. We observe many false alarms if the SVM alone is used for the lane change intention prediction. Tab. I gives the performance results in terms of the average precision, recall and prediction time for the SVM and SVM+BF approaches. We can see from the table that the precision is improved from 0.2857 to 0.7154 by adding the Bayesian Filter. The average prediction time is decreased by 0.022 seconds, which is acceptable with such a remarkable improvement on the precision. Fig. 6 displays the histogram of prediction time for 54 lane changes. Most of the lane changes are predicted almost 1.3 seconds before the ego vehicle crosses the painted line and that the maximum prediction horizon reaches 3.29 seconds.
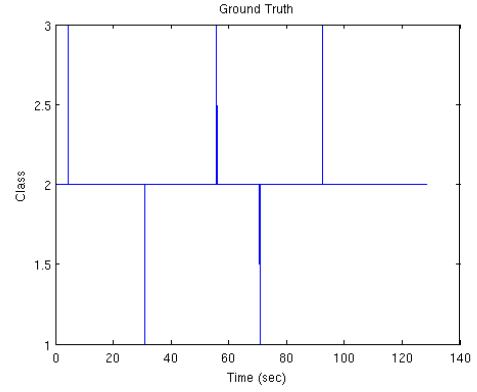
TABLE I
COMPARISON OF SVM AND SVM+BF (TESTED ON 69 LANE CHANGES)

Cases	Precision	Recall	Avg prediction time (sec)
SVM	0.2857	1	1.2947
SVM + BF	0.7154	1	1.2718

C. Finding the best parameters

As mentioned earlier there are mainly five parameters involved during the implementation of the proposed approach (SVM+BF) and each parameter has a strong effect on the performance. These parameters are: g, C, t1, t2, and f. Simultaneously finding the best parameters is a five dimensional grid search problem and since the values of each parameter may vary in a large range it is computationally very expensive to do. Instead a simplified strategy is used:

1) Fix $t1$, $t2$, f and find the best C and g using grid search with exponentially growing sequences of C and g [17], for example $C = (2^{-3}, 2^{-1}, \ldots, 2^3)$ and $g = (2^{-4}, 2^{-2}, \ldots, 2^2)$. While fixing $t1$, $t2$ and f it was made sure that these parameters gave good results when tested on different scenarios.
2) Fix C and g to their best values, fix $t1$ and $t2$, and find the best value for f.
3) Fix C, g, and f to their best values and then iterate over a range of $t1$ and $t2$ to find their best values.

Fig. 6 shows performances for different values of the parameters. The plots show that choosing bad parameters can lead to situations with very poor precision or recall.

D. Robustness evaluation

To test the effect of different drivers on the performance of the proposed approach, the following cases were considered with our two drivers x and y:

- Case 1: Training with x and testing with y.
- Case 2: Training with y and testing with x.
- Cases 3(a) and 3(b): Training with x and testing with different driving scenarios with x.

Tab. II shows the performance for each case. The recall is always 1. The F-1 score, precision and prediction time are almost the same, which shows the robustness of the system.
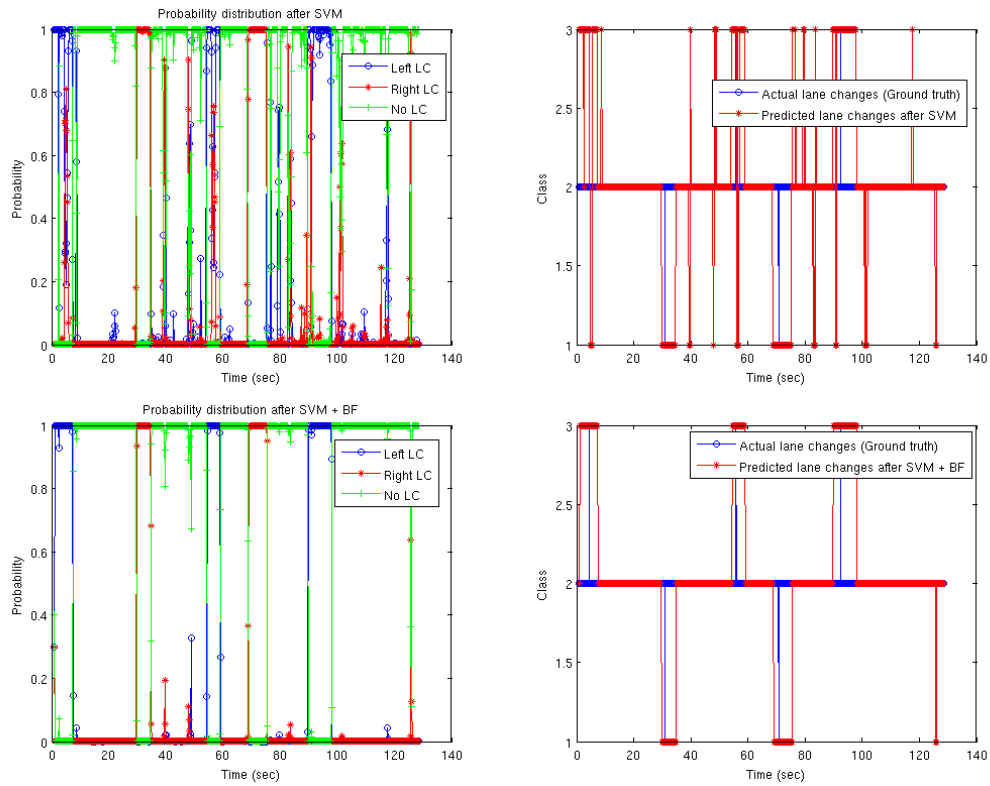
Fig. 5. Comparison of lane change prediction using SVM and SVM+BF. (Top Left + Top Right): Probability distribution for all the three classes using SVM, and lane change prediction based on this distribution. (Bottom Left + Bottom Right): Probability distribution for all the three classes using SVM+BF, and lane change prediction based on this distribution.
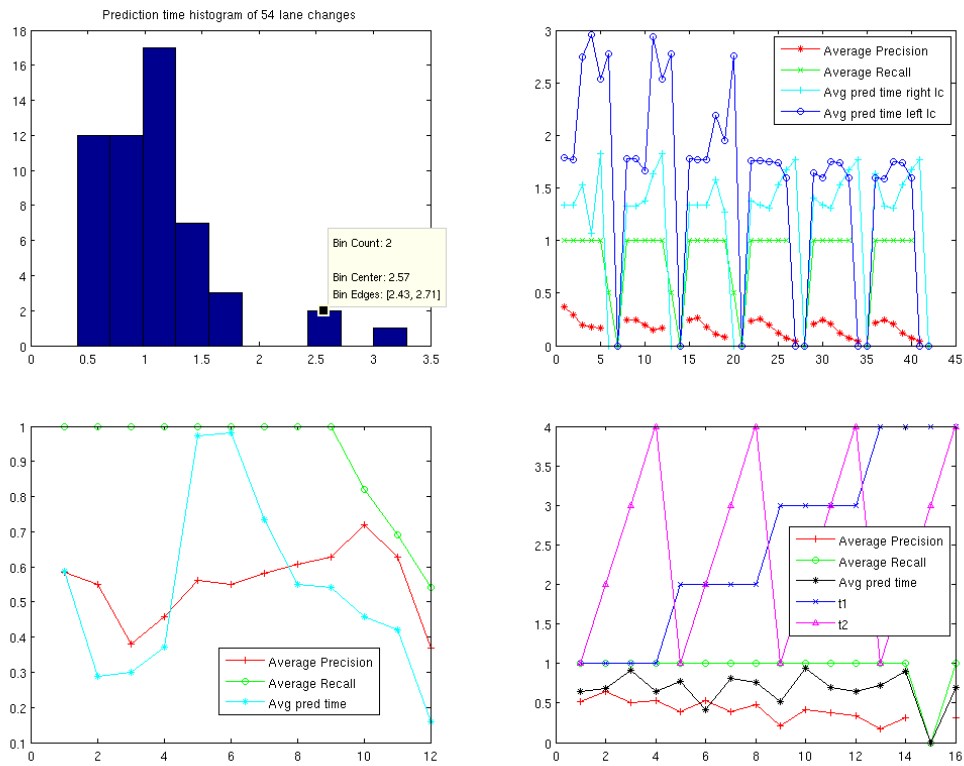

Fig. 6. (Top Left): Histogram of prediction time. (Top Right): Plot of precision, recall and average prediction time for left and right lane changes for the selection of the best (C,g). (Bottom Left): average precision, recall and prediction time for finding the best f. (Bottom Right): average precision, recall and prediction time for finding the best (t1, t2). Time is expressed in seconds.

Cases	Precision	Recall	F1-Score	Avg. Prediction Time (secs)
Case 1	0.7849	1	0.8717	1.0032
Case 2	0.55	1	0.7084	0.9495
Case 3(a)	0.8235	1	0.9032	0.9711
Case 3(b)	0.6465	1	0.7786	1.0687

V. CONCLUSIONS

This paper proposed a solution to lane change intention prediction based on a combination of a multiclass SVM classifier and Bayesian filtering. When tested on real data, the algorithm is able to predict the lane changes accurately (recall = 1) on average 1.3 seconds before they occur and to differentiate between left lane changes and right lane changes. Future work will focus on the reduction of the number of false alarms. In our current system most of the false alarms are caused by inaccuracies of the lane tracker, which sometimes fluctuates and jumps. Therefore in the future the lane tracker should be improved. Further improvements could be obtained by incorporating additional information to our prediction framework, such as the distance to the vehicle in front or the speed difference with the vehicle in front. Indeed, these are useful cues to estimate whether the driver intends to overtake the vehicle in front. Also an extended evaluation with a larger dataset and several drivers is to be done.

REFERENCES

[1] H. Lum and J. A. Reagan, "Interactive highway safety design model: accident predictive module," *Public Roads*, vol. 59, no. 2, 1995.

[2] A. Boubezoul, A. Koita, and D. Daucher, "Vehicle trajectories classification using support vectors machines for failure trajectory prediction," in *Proc. IEEE International Conference on Advances in Computational Tools for Engineering Applications*, pp. 486–491, 2009.

[3] H. M. Mandalia and D. D. Salvucci, "Using support vector machines for lane change detection," in *Proc. of the Human Factors and Ergonomics Society 49th Annual Meeting*, 2005.

[4] J. C. McCall, D. P. Wipf, M. M. Trivedi, and B. D. Rao, "Lane change intent analysis using robust operators and sparse bayesian learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 431–440, 2007.

[5] B. Morris, A. Doshi, and M. M. Trivedi, "Lane change intent prediction for driver assistance: on-road design and evaluation," in *Proc. IEEE Intelligent Vehicles Symposium*, pp. 895–901, 2011.

[6] G. Aoude, V. Desaraju, L. H. Stephens, and J. P. How, "Behavior classification algorithms at intersections and validation using naturalistic data," in *Proc. IEEE Intelligent Vehicles Symposium*, pp. 601–606, 2011.

[7] N. Oliver and A. P. Pentland, "Graphical models for driver behavior recognition in a smartcar," in *Proc. IEEE Intelligent Vehicles Symposium*, pp. 7–12, 2000.

[8] D. Meyer-Delius, C. Plagemann, and W. Burgard, "Probabilistic situation recognition for vehicular traffic scenarios," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 459–464, 2009.

[9] C. Tay, *Analysis of dynamic scenes: application to driving assistance*. PhD thesis, Institut National Polytechnique de Grenoble, France, 2009.

[10] T. Hülnhagen, I. Dengler, A. Tamke, T. Dang, and G. Breuel, "Maneuver recognition using probabilistic finite-state machines and fuzzy logic," in *Proc. IEEE Intelligent Vehicles Symposium*, pp. 65–70, 2010.

[11] D. Wipf and B. Rao, "Driver intent inference annual report," tech. rep., University of California, San Diego, San Diego, 2003.

[12] G. S. Aoude, *Threat assessment for safe navigation in environments with uncertainty in predictability*. PhD thesis, Massachusetts Institute of Technology, USA, 2011.

[13] T. K. Huang, R. C. Weng, C. J. Lin, and G. Ridgeway, "Generalized bradley-terry models and multi-class probability estimates," *Journal of Machine Learning Research*, vol. 7, pp. 85–115, 2006.

[14] J. Goldbeck and B. Huertgen, "Lane detection and tracking by video sensors," in *Proc. IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, pp. 74 –79, 1999.

[15] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[16] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proc. International Conference on Machine Learning*, pp. 104–112, 2004.

[17] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," tech. rep., National Taiwan University, Taipei, 2003.