2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)

Vehicle Trajectory and Lane Change Prediction Using ANN and SVM Classifiers

R. Izquierdo, I. Parra, J. Muñoz-Bulnes, D. Fernández-Llorca and M. A. Sotelo

Abstract— **Millions of traffic accidents take place every year on roads around the world. Some advanced assistance systems have been released in commercial vehicles in the past few years, contributing to the transition towards semiautonomous vehicles. Some of the best known are the adaptive cruise control and the lane keeping systems. These systems keep a desired distance with respect to the preceding vehicle or a fixed speed on the center of the lane. It is very useful for these systems to know what the surrounding vehicles trajectories will be or if they will perform a lane change manoeuvre. This paper evaluates two kinds of artificial neural networks over two different datasets to predict its trajectories. A Support Vector Machine classifier is used to classify the action that will be carried out. The proposed trajectory prediction systems are 30% better than the vehicle motion model in a time horizon of 4 seconds and are able to predict a lane change action 3 seconds before it happens.**

I. Introduction and Related Works

More than one million accidents occurred in the European Union roads in 2014 according to the *Annual Accident Report 2016* published by the European Road Safety Observatory. About 26.000 of these accidents were fatalities and 62% occurred outside urban areas. According to the *Global Status Report on Road Safety* published by the World Health Organization in 2015 these numbers rise to 1.2 million fatalities in 2013 in road traffic accidents.

Manufacturers have been introducing in their vehicles some advanced driver assistance systems with a view to reduce the high number of accidents during the last few years. Systems such as Adaptive Cruise Control or Collision Avoidance Systems are currently working on the vehicles. In highway scenarios it is especially interesting for the ADAS be able to predict the trajectory of the surrounding vehicles to take actions into account in order to reduce the risk's situation or to improve a more comfortable or efficiency drive. In the same way, the lane change prediction is also interesting to predict a cut-in or cut-off manoeuvre.

A good deal of research has already been done in order to predict the trajectories or the lane change intention. For example, in [1] the trajectories have been transformed into a Quaternion-based Rotationally Invariant framework to classify the trajectory. Predictions are made based on this classification. In [2], [3] an Artificial Neural Network (ANN) is used to predict the lateral motion of the vehicle over the NGSIM dataset [4]. A manoeuvre recognition based on kinematic measures and road geometry is proposed in [5], the Constant Yaw Rate Acceleration model and the detected manoeuvre are used as input of the trajectory generation system. The results showed indicates a mean error below 0.45 meters up to 4 seconds. In [6] a lane change trajectory is performed searching the k-nearest real lane change in a database that best match with the evaluated road scene.

On the other hand, many other works have focused on predicting when a lane change will happen. Thus, in [7] a Support Vector Machine and a Bayesian filter are used to predict the lane change taking into account the lateral position and the heading error of the vehicle with respect to the road. As is shown in [8], the most discriminating features when predicting a lane change are the lateral speed, the preceding vehicle's speed and the lateral position with respect to the lane centre. Using these inputs in a Naïve Bayes algorithm, lane changes are detected precisely up to 2.2 seconds in advance. In [3] a desired lane is predicted by an ANN to asses and predict the lane change when the desired lane is different from the current lane. This lane change can be predicted up to 3.5 seconds before it actually happens. The trajectories predicted can also be used to detect a lane change. In [5] the classification is evaluated too. The lane change is detected 1.15 seconds before it happens corresponding with a mean value of 0.33 meters before crossing the lane.

Recently some databases have been made available to the scientific community in the vehicle's motion prediction area, setting the basis to compare results and making this type of research more accessible. One of the best-known databases is the NGSIM and the other is the University of Peking database[9]. The last one is little known but it has been recorded over the mobile platform perspective. This database is used in this paper and in [10], [11].

This paper evaluates the best configuration for two kinds of neural networks predicting the trajectory of the vehicle a few seconds ahead over two different databases. An SVM classifier is used in a second stage in order to predict whether a lane change will occur (a few seconds before it really happens).

The rest of the paper is organized as given: Section II describes the dataset used in this paper. The features extraction is detailed in Section III. Section IV describes the lateral motion prediction system. The lane change detection system is presented in Section V. In Section VI some experimental results are presented. The conclusions and the future works are described in Section VII.

R. Izquierdo, I. Parra, J. Muñoz-Bulnes, D. Fernández-Llorca and M. A. Sotelo are with the Computer Engineering Department, University of Alcalá, Alcalá de Henares, Spain `ruben.izquierdo@uah.es`

978-1-5386-1526-3/17/$31.00 ©2017 IEEE

II. Dataset Description

The goal of this paper is to evaluate the performance of two kinds of ANNs predicting the lateral motion of the ego vehicle. It is realistic to assume that the V2V communications will be implemented in new cars in a few years. These communication systems can do this prediction extensible to the surrounding vehicles. These two ANN architectures are evaluated in two datasets in order to achieve different results in different datasets with different variability. The first dataset used in this paper was published by the University of Peking [9] and the other one has been developed by the University of Alcalá. Table I shows the characteristics of each dataset.

A. University of Peking Dataset

The dataset published by the University of Peking contains data recorded in the 4th ring road in Peking during 97 minutes and more than 69 km. The mobile platform has a GPS-IMU system to perform the ego position in a global reference system and 4 Lidar sensors to compute the surrounding vehicles position and the road limits coordinates in the ego and global reference systems. Hereafter this dataset will be named as the PKU dataset. The data is not hard-synchronized. The ego data has a rate close to 20 Hz and the surrounding's vehicles data is about 10 Hz. The data has been synchronized with a fixed time stamp corresponding with a rate of 10 Hz, fitting a 3rd order polynomial with one second of the past and one second of the future of the data to be fitted.

B. University of Alcalá Dataset

The dataset recorded by the University of Alcalá consists of more than 60 minutes and 100 km driven along a High Accurate Digital Map (HADM) at the E-90 highway between Alcalá de Henares and Guadalajara. The vehicle position was recorded with a GPS at fixed rate of 10 Hz and fused with the odometry provided by the CANBUS using an EKF. The positions of the surrounding vehicles were not recorded in this dataset. Hereafter this dataset will be named as UAH dataset.

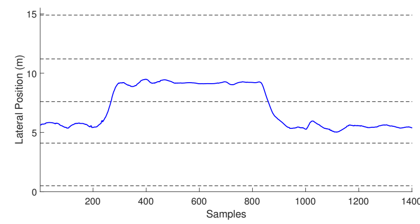
	Length	Distance	Lane Changes	Mean Lane Change Time
PKU	97 min	69 km	165	6.8 s
UAH	60 min	100 km	186	5.9 s

TABLE I

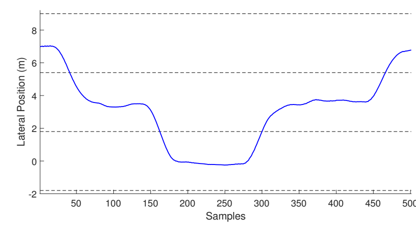
Datasets Main Characteristic

III. Features extraction

The main input feature to predict the lateral position of the vehicle with respect to the road reference system is the temporal evolution of the vehicle lateral position. The left boundary has been used as zero reference level to compute this value in the PKU dataset. The left boundary corresponds with a continuous median parallel and equidistant to the road. This wall allows to compute a continuous lateral position invariant to the road path which represents the trajectory of the vehicle in the road reference system. However, the left boundary detection is a little bit noisy, in order to reduce it a polynomial RANSAC filter has been performed over the raw boundary data. The lateral position of the vehicle is computed with respect to the centre of the left lane contained in the HADM in the UAH dataset. The HADM consists of a set of GPS points which represent the lane centre positions of each lane along the road path. The computed lateral position in the road reference system is shown in Figure 1.



(a)



(b)

Fig. 1. Main Feature: lateral position in (a) PKU (b) UAH datasets.

The second feature is the heading error, defined as the vehicle heading in the road reference system. This has a zero value when the vehicle is aligned with the road direction. The last feature used is the lateral speed defined as the value difference of the lateral position between two samples or computed using the heading error and the vehicle speed. The features extracted from the PKU and the UAH data set are shown in Figure 2. An evaluation of the lateral speed feature in the PKU dataset reveals that it is noisier than the UAH dataset. It is computed based on the lateral position. Then the lateral position on the PKU dataset is also noisier than in the UAH dataset.

A. Baseline Performace

All the datasets are different and the results performed with them have to be relativized. Many researches refer theirs results with respect to the Constant Yaw Rate Acceleration (CYRA) model [12], [13]. However, this method implies some degrees of freedom that cannot be met in a road reference system. In order to provide a quantitative measure of the difficulty to predict the lateral position of the vehicle in each dataset a baseline score has been computed. The baseline scores are the mean absolute error expressed in meters for two cases: Constant Lateral Position (CLP) or Constant Heading Direction (CHD). Table II shows this baseline results for both datasets at different time horizon predictions. As can be seen, the UAH dataset is harder to predict than the PKUs.

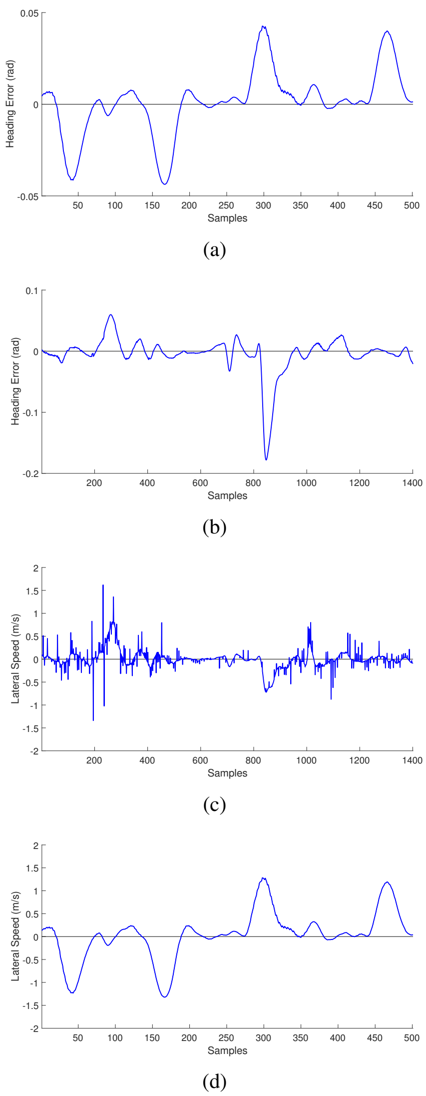


Fig. 2. Secondary Features: (a) and (b) heading error, (c) and (d) lateral speed in PKU and UAH datasets respectively.

Prediction Time (seconds)	1	2	3	4
PKU - CLP	0.18	0.33	**0.45**	**0.56**
PKU - CHD	**0.12**	**0.28**	0.48	0.71
UAH - CLP	0.31	0.57	0.82	**1.06**
UAH - CHD	**0.14**	**0.41**	**0.76**	1.16

TABLE II
BASELINE RESULTS

IV. LATERAL MOTION PREDICTION SYSTEM

Two different architectures of neural networks have been used in order to perform the lateral position prediction. The first one is a Nonlinear Autoregressive Neural Network (NARNN) which is especially indicated to predict time series in dynamical models. The second one is a feed-forward neural network (FFNN) which is indicated when a mapping between inputs and outputs is desired.

A. Training setup

The dataset has been divided in three subsets for the training process in blocks of 2, 1 and 1 thousand samples continuously built to avoid overlaps. The training, validation and test subsets represent 50%, 25% and 25% respectively. The weights and bias optimization method used in the training is the Levenberg-Marquardt algorithm [14], [15] with early stopping. The training subset is used to update the weights during the training process, the validation subset is used to stop the training process when the error starts to rise and the test subset has no effect during the training process. It is only used to compute the training score.

B. NARNN Parameters

The NARNN uses as input the same variable as it tries to predict. In this case, the lateral position at a given rate by the datasets. Note that this type of ANN predicts only one step ahead as shown in eq. 1. To perform the prediction more than one step ahead, the current predicted output is transformed into an input value. The second step will be performed with one less real input data using instead the predicted values as shown in eq. 2.

$$\hat{y}_{t+1} = f\left(y_t, \cdots, y_{t-d}\right) \tag{1}$$

$$\hat{y}_{t+2} = f\left(\hat{y}_{t+1}, y_t, \cdots, y_{t-d+1}\right) \tag{2}$$

Several tests have been done with the NARNN's to achieve the best parameters setup. The input delays d, the number of hidden layers and the hidden layer size has been tested in this kind of NN. The hidden and output layers's neurons have sigmoid and linear transfer function respectively. Table III shows the three best results for the following configuration sets, $d = \{1, 10, 20, 30, 40\}$, $nhl = \{1, 2\}$ and $hls = \{10, 20, 50, 100\}$.

Delays	**Hidden Layer Size**	**1s**	**2s**	**3s**	**4s**
PKU Dataset					
20	20	0.13	0.28	0.43	0.55
1	10	0.20	0.36	0.49	0.62
30	20	0.13	0.28	0.42	0.57
UAH Dataset					
40	10-10	0.29	0.58	0.88	1.17
30	20	0.31	0.62	0.93	1.24
30	10-10	0.30	0.59	0.92	1.25

TABLE III
NARNN PREDICTION ERRORS

C. FFNN

The FFNN uses as input whatever variable to compute other values that could be the same as the input or not. Eq. 3 shows the features vector F at time t, where y represent the lateral position, θ the heading and $\dot{y}$ the lateral speed.

Equation 4 shows the input vector X with different window sizes d and finally eq. 5 shows the output target vector where k represents the data rate which is 10 in both databases, being T the lateral position at $t = t + \{1,2,3,4\}$ seconds.

$$F_t = [y_t, \theta_t, \dot{y}_t] \tag{3}$$

$$X_t = [F_t, \cdots, F_{t-d}] \tag{4}$$

$$T_t = [y_{t+k}, \cdots, y_{t+4k}] \tag{5}$$

To obtain the best result with this ANN architecture, different configurations have been tested. The number of hidden layers, the size of the hidden layers and the number of inputs have been taking into account. The hidden layers and the output neurons have a sigmoid and linear transfer function respectively. The length of the feature vector, starting with one and finishing with three elements, and its number of samples in the past d have also been tested. The results of the three best configurations are shown in Table IV. The following sets has been tested $d = 20$, $nhl = \{1,2\}$, $hls = \{10,20,50,100\}$ and $F = \{1, 1:2, 1:3\}$.

Features	Hidden Layer Size	1s	2s	3s	4s
PKU Dataset					
y	20	0.10	0.23	0.35	0.42
$y\ \theta$	10-10	0.10	0.21	0.33	0.45
$y\ \theta\ \dot{y}$	50	0.10	0.21	0.32	0.44
UAH Dataset					
y	50-50	0.11	0.29	0.51	0.73
$y\ \theta$	10	0.10	0.27	0.49	0.71
$y\ \theta\ \dot{y}$	10	0.09	0.26	0.48	0.70

TABLE IV
FFNN PREDICTION ERRORS

V. LANE CHANGE PREDICTION SYSTEM

There are two different but related tasks in the lateral motion prediction. The first one is to predict the vehicle trajectory and the second one is to predict if a lane change will happen. This task consists of the classification of the future vehicle's manoeuvre. In order to predict if a lane change will happen a well-known SVM classifier [16] with a Radial Basis function as kernel has been used. The inputs and the labelling used to train the SVM classifier are detailed below.

A. Classification Inputs

The three features, y, θ and $\dot{y}$ have been used to make the classification. The input vector is built with the last two seconds of features. With a given 10 Hz rate the input vector has 60 elements. The predicted positions made by the NN's have been included in the input vector to improve the classification results. Eq 6 shows the input vector.

$$X_t = [F_t, \cdots, F_{t-20}, T_t] \tag{6}$$

B. Lane Change Labelling

It is very important to make a correct labelling that allows the SVM learn when a lane change is happening. The Lane Change Point (LCP) is defined by the crossing of the vehicle's centre over the line between two lanes (lane marking). In the UAH dataset the mean lane change time is over six seconds symmetrically split before and after the LCP. The prediction time desired to evaluate is four seconds, consequently four seconds before and three seconds after the LCP have been labelled, as is depicted in Figure 3.

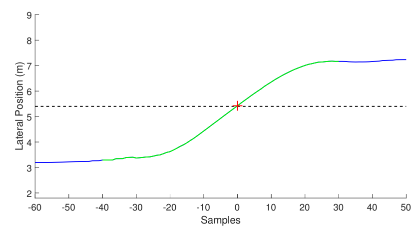


Fig. 3. Lane change labelling. Blue line represents the vehicle's trajectory, green line the trajectory subsequence labelled as lane change and the red square the lane change point.

For the training, two thirds of the UAH dataset have been used as training subset and the other third has been used to testing the classifier. Figure 4 shows the Receiver Operating Characteristic (ROC) for the training and test subsets. When the time prediction horizon becomes bigger the Area Under the Curve (AUC) becomes smaller. Table V shows numeric results for the classification process.

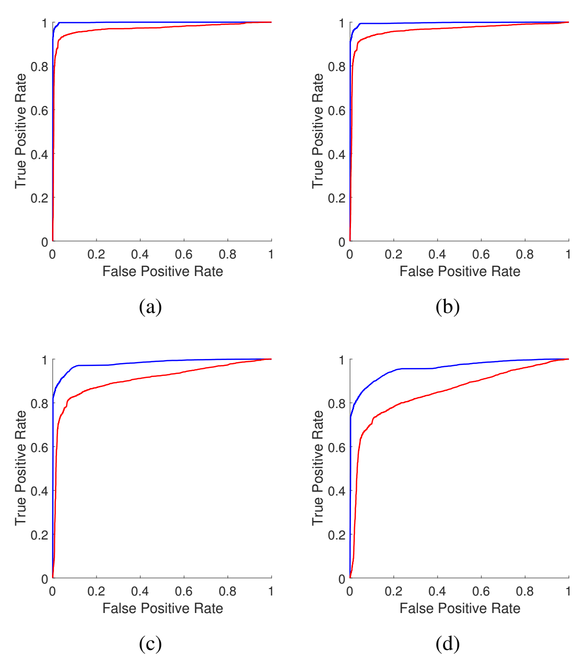


Fig. 4. Receiver Operating Characteristic for: (a) 1 sec. (b) 2 sec. (c) 3 sec. (d) 4 sec. prediction time. Blue line represents the results over the training set and red line over the test set.

Pred. Time	1 s	2 s	3 s	4 s
TPR	0.92	0.90	0.81	0.73
FPR	0.03	0.03	0.07	0.11
F1-Score	0.93	0.92	0.85	0.79

TABLE V

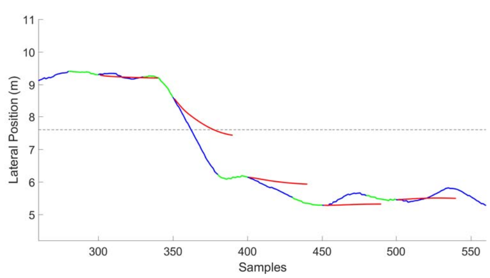
SVM CLASSIFICATION SCORES

VI. RESULTS

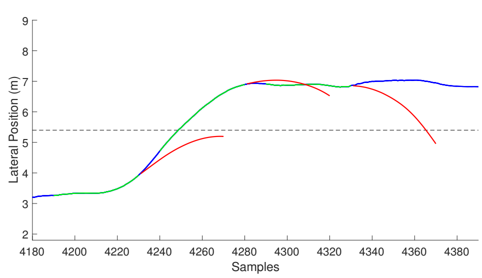
In this section some results for the trajectory estimation and classification are given.

A. *ANN trajectories prediction*

Firstly, the trajectory estimated for the NARNN is shown over the two datasets. A short sequence is shown in Figure 5. In this sequence three actions are carried out. First a lane keeping, followed by a full lane change, and finally lane keeping again. This prediction method is not better than the baseline method. The NARNN learns one step ahead so the errors in a short prediction time are smaller but not in longer step predictions. The results obtained with this method look like the CYRA model because the ANN has learned the vehicle motion model instead of learning the lane keeping and lane change dynamics. The training process tries to assess a low error with one step prediction. However the prediction made with it a few steps ahead has not been trained.



(a)

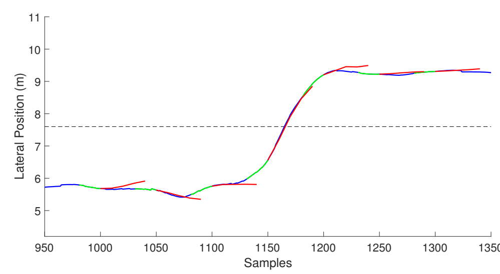


(b)

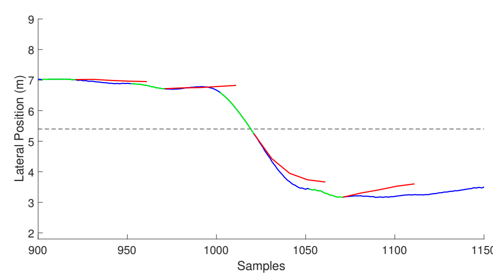
Fig. 5. Lateral trajectories prediction: (a) NARNN on PKU dataset (b) NARNN on UAH dataset. Blue line is the real trajectory, green represents the input data for the predicted values in red.

The trajectories predicted by the FFNN are shown in Figure 6 for both datasets. A lane keeping, lane change and lane keeping manoeuvre are carried out during the sequence. When the lane keeping is made the system predicts a non-accurate trajectory which does not represent a lane change. This is produced by the wavy movement produced by the driver while is keeping a constant reference. The prediction system becomes more accurate during the lane change. In both figures 6(a) and 6(b) a prediction is made just when the vehicle crosses or has crossed the line. In this point the lane change process is fully developed and the prediction system is able to predict the final lateral position.

This type of ANN has been trained to learn the vehicle position a few seconds ahead directly, instead of one step ahead. This fact makes the prediction better with respect to the NARNN, the CLP or the CHD methods as we show in the section IV-C. The prediction error falls 23% with respect to the baseline errors in the PKU dataset and a 30% in the UAH dataset. As can be seen in Table II, the baseline errors are bigger in the UAH dataset. This means that the difficulty to predict future positions is bigger, but it is where the FFNN is better. This can be explained by the low noise of the UAH dataset.



(a)



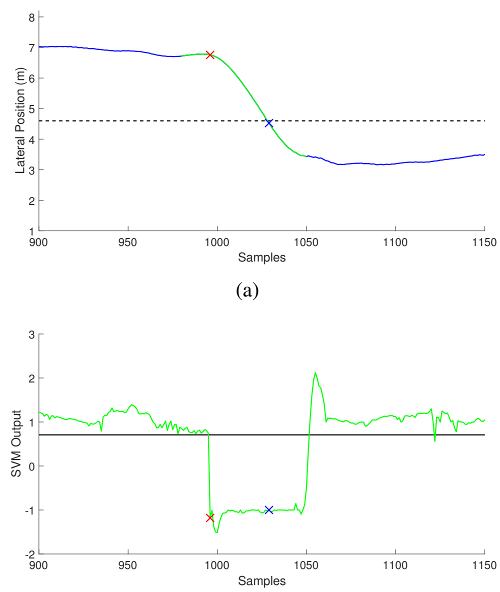
(b)

Fig. 6. Lateral trajectories prediction: (a) FFNN on PKU dataset, (b) FFNN on UAH dataset. Blue line is the real trajectory, green represents the input data for the predicted values in red.

B. *SVM Manoeuvre Classiffication*

The SVM classifier is able to detect the lane change up to four seconds before it happens, as far as the prediction time in the detection probabilities falls. The manoeuvre detection process is illustrated in Figure 7. In Figure 7(a), the trajectory followed by the vehicles is shown in blue while the sub trajectory labelled as a lane change is plotted in green. Blue marker represents the lane change event and the red one represents the event detection. The detection trigger is a level comparison, which matches with the value that maximizes the F1-Score. The manoeuvre recognition has been triggered

3.3 seconds (33 samples) before the lane change happens. At sample 1122 a false positive is detected which is an isolated value. Many false positive can be removed adding some delays and a low-pass filter to the decision making.



(a)

(b)

Fig. 7. SVM Manoeuvre recognition. (a) In blue real trajectory, green sub trajectory labelled as lane change. (b) SVM output in green and the trigger level to make the classification in black. Red and blue markers represent lane change detection and lane change event respectively.

VII. Conclusions and Future Works

A baseline method has been proposed to evaluate the performance of the lateral position prediction algorithms. These baseline scores have been computed for the PKU and UAH datasets, and two kinds of NN have been tested with them. The NARNN, especially indicated to predict dynamical systems, is not better than the baseline method, however the FFNN is able to reduce the mean absolute error on the predicted positions about 23% and 30% up to 4 seconds in the PKU and UAH datasets respectively. For the lane change detection, a SVM classifier has been tested. The SVM can detect precisely a lane change 3 seconds before it happens.

The presented results are based on kinematics and road structure data only, for this reason, the predicted values of the position can be well predicted only when evidence of those exists. Adding context information could be useful to improve the performance of the prediction systems. More powerful machine learning methods such as CNNs can also improve the performance. Wavy movement has been observed in a lot of sequences in the database when a lane keeping is carried out. A frequency domain analysis could provide discriminative features between the lane keeping and lane change actions. The SVM classification output can be easily improved by implementing a Bayesian Filter or a Hidden Markov Model [17].

VIII. Acknowledgements

This work was supported by Research Grants SEG-VAUTO S2013/MIT-2713 (Autonomous Community of Madrid), DPI2014-59276-R (Spanish Ministry of Economy), FPU14/02694 (Spanish Ministry of Science, Culture and Sports) via a predoctoral grant to the first author and BRAVE H2020 GA 723021 (European Commission). This project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737469. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Austria, Spain, Italy, Latvia, Belgium, Netherlands, Sweden, Finland, Lithuania, Czech Republic, Romania, Norway.

References

[1] C. Hermes, C. Wohler, K. Schenk, and F. Kummert, "Long-term vehicle motion prediction," in *2009 IEEE Intelligent Vehicles Symposium*, June 2009, pp. 652–657.

[2] R. S. Tomar, S. Verma, and G. S. Tomar, "Prediction of lane change trajectories through neural network," in *2010 International Conference on Computational Intelligence and Communication Networks*, Nov 2010, pp. 249–253.

[3] S. Yoon and D. Kum, "The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 1307–1312.

[4] F. H. Administration, "The next generation simulation program (ngsim)," http://ngsim-community.org/.

[5] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 4363–4369.

[6] W. Yao, H. Zhao, P. Bonnifait, and H. Zha, "Lane change trajectory prediction by using recorded human driving data," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, June 2013, pp. 430–436.

[7] P. Kumar, M. Perrollaz, S. Lefvre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, June 2013, pp. 797–802.

[8] J. Schlechtriemen, A. Wedel, J. Hillenbrand, G. Breuel, and K. D. Kuhnert, "A lane change detection approach using feature ranking with maximized predictive power," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, June 2014, pp. 108–114.

[9] U. of Peking, "University of peking database (pku)," http://poss.pku.edu.cn/download/.

[10] H. Zhao, C. Wang, Y. Lin, F. Guillemard, S. Geronimi, and F. Aioun, "On-road vehicle trajectory collection and scene-based lane change analysis: Part i," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 192–205, Jan 2017.

[11] W. Yao, Q. Zeng, Y. Lin, D. Xu, H. Zhao, F. Guillemard, S. Geronimi, and F. Aioun, "On-road vehicle trajectory collection and scene-based lane change analysis: Part ii," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 206–220, Jan 2017.

[12] A. Berthelot, A. Tamke, T. Dang, and G. Breuel, "Handling uncertainties in criticality assessment," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 571–576.

[13] A. Tamke, T. Dang, and G. Breuel, "A flexible method for criticality assessment in driver assistance systems," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 697–702.

[14] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[15] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[16] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[17] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.