# An Effective Driver Intention and Trajectory Prediction for Autonomous Vehicle based on LSTM

Fatimetou El Jili
*Altran Prototypes Automobiles, Research Departement, 02 rue Paul Dautier, 78140, France*

Keywords: Artificial Intelligence, Intention Prediction, Trajectory Prediction, Deep Learning, Long Short Term Memory, CARLA Simulator, Autonomous Driving.

Abstract: In order to make the navigation system of autonomous vehicle more robust and safe in urban environment we propose in this paper a model for driver intention prediction and trajectory prediction. The proposed model is based on LSTM (long short term memory). The model was trained on database of features collected from the driving simulator CARLA. This paper treats four type of intentions, turn left, turn right, go straight and stopping intention. Two cases were treated, the first case is to predict intention before it occurs, the second case corresponds to intention recognition, where the driver already starts maneuvering the intention. Both cases are treated by the same model. The model shows better performances for the second case than the first case with small differences. The main strength of our model is that it gives good performances with a small set of features. The accuracy of the model is 96% for intention prediction and 97% for the intention recognition. The proposed method for trajectory prediction reach an accuracy of 99.9%. Those accuracies are higher than what we found in state of art.

## 1 INTRODUCTION

Autonomous driven is a very complex system that requires a lot of constraints to perform as the best human driving or better. In order to make a robust and safe navigation system, understanding other driver's intentions is one of the most important task. By predicting the surrounding vehicles intentions, the autonomous vehicle can plan its trajectory in a way that it can avoid collision with other vehicles. In general collisions happen due to a false identification of driver's intentions or a lack of attention from the driver.

In this paper we propose a method for driver intention prediction and recognition for self driving vehicles at several type of intersection (tree way, two way and four way intersections), where intentions are turn left and turn right, go straight and, stop, it also gives intention on one way where in ideal case intentions are stopping or going straight. This method also gives the direction the vehicle is fellowing to manoeuvre the intention. We also propose a method for trajectory prediction. These methods can be used for ADAS systems.

More recently in the last decade with the apparition of autonomous vehicle and ADAS system, driver intention prediction has been a topic of interest of researchers. A variety of approches were proposed for driver intention prediction. Some statistical methods (L.R Rabiner, 1986), (Streubel and Hoffmann, 2014), (Hou and al., 2011) were proposed to solve this problem. Some machine learning methods like SVM (Support Vector Machines) in (B. Tang, 2015) and GP (Gaussian Process) (Laugier and al., 2011) were also used to solve this problematic. Most of these work needs a huge datasets to train their model and some complex features, like when lane detection is required, these additional tasks related to those features computational time give rise to the model computational time increasement. Whereas, our model use a small dataset and a small set of features which doesn't need any additional tasks or complex artificial intelligence algorithms to compute those features.

In (Hou and al., 2011) authors used CHMM (Continuous Hidden Markov Model ) for driver intention prediction which gives an accuracy of 95% for intention recognition, while in (B. Tang, 2015) SVM gives an accuracy of 90% 1.6 s before the intersection and an accuracy of 93% at intersection for a generalized method for driver intention prediction at intersection. There are other approches based on deep learning specially RNN (Recurrent Neural Network) (A. Zyner and Nebot, 2018), LSTM (Long-Short Term Memory) (Sepp Hochreiter, 1997), (Hao Xue



El Jili, F.
An Effective Driver Intention and Trajectory Prediction for Autonomous Vehicle based on LSTM.
DOI: 10.5220/0010321710901096
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 1090-1096
ISBN: 978-989-758-484-8

and Reynolds, 2018), (Derek J. Phillips and Kochenderfe, 2017), deep inverse reinforcement learning (Zhang and al., 2018) and deep convolutional network in (Djuric and al., 2019). In (Derek J. Phillips and Kochenderfe, 2017) the LSTM model gives 95% on dataset regrouping all kind of intersections. In (B. Tang, 2015) authors proposed a method based on HMM (Hidden Markov Model) for prediction of driver intended path which gives an accuracy up to 90% 7 seconds before entering the intersection area.

Our model gives a higher accuracy than what we found in state of art. This model doesn't require any information about roads and the map. Section 2 illustrates the LTSM(Sepp Hochreiter, 1997) model used in this work to perform the prediction. In section 3 we present an overview of the proposed method. Section 4 exhibits the model selection and datasets collection details.

## 2 LSTM: LONG SHORT TERM MEMORY

LSTM (Long Short Term Memory) (Sepp Hochreiter, 1997) is a novel architecture of recurrent neural network (A. Zyner and Nebot, 2018) with an appropriate gradient based learning algorithm. The RNN (Recurrent Neural Network) can use past information when the time gap between past and present is short, whereas when the time gap become long the RNN can not learn exact information from the past.LSTM was designed to remediate this problem so it can learn from the past even when the time gap between past and present is long, it also can learn when the input data is incomprehensible due to noise. It is useful for sequential data, time series data, speech processing, etc ...

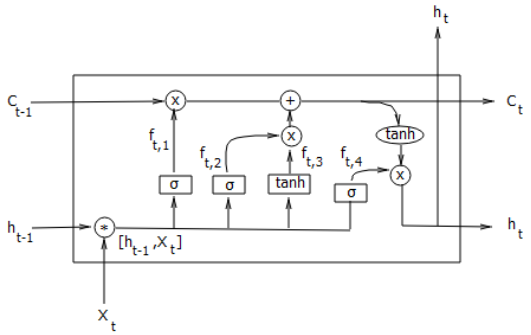


Figure 1: LSTM variant.

Figure 1 illustrate an exemple of LSTM variant at step $t$ of the model, or in other words when the input of the model is the sequence of data collected at time $t$. There is many variants of LSTM, researchers have shown that almost all variants have the same performances. The variables $X_t$, $C_{t-1}$ and $h_{t-1}$ are respectively the input of the network at the step $t$, the memory cell at the step $t-1$ and the output of the network at the step $t-1$. These variables are given as the input of the network at the step $t$ to compute the output $h_t$ and the memory cell $C_t$. The memory cells stores the information about the past at each step of the model in order to be used as input of the next step. For the variant in figure 1, the memory cell $C_t$ and the output $h_t$ are given by the following equations:

$$C_t = f_{t,1}C_{t-1} + f_{t,2}f_{t,3}, \quad (1)$$

$$h_t = f_{t,4}tanh(C_t), \quad (2)$$

Where $f_{t,1}$, $f_{t,2}$, $f_{t,3}$ and $f_{t,4}$ are given by the equations below :

$$f_{t,1} = \sigma(W_{t,1}[h_{t-1}, X_t] + B_{t,1}), \quad (3)$$

$$f_{t,2} = \sigma(W_{t,2}[h_{t-1}, X_t] + B_{t,2}), \quad (4)$$

$$f_{t,3} = \tanh(W_{t,3}[h_{t-1}, X_t] + B_{t,3}), \quad (5)$$

$$f_{t,4} = \sigma(W_{t,4}[h_{t-1}, X_t] + B_{t,4}), \quad (6)$$

$B_{t,i}$ and $W_{t,i}$, $1 \leq i \leq 4$, correspond respectively to the biais vector and the weight matrix, and $\sigma$ the softmax function.

## 3 THE PROPOSED METHOD FOR DRIVER INTENTION AND TRAJECTORY PREDICTION

This paper focus mainly on intention prediction, recognition, and trajectory prediction. The term recognition is used when the intention already occurred or its manoeuvre already starts. In this work intention prediction is treated as a classification problem. Given the past information : starting from the present back to the past though a given interval of time, we predict or recognize the driver intention. Trajectory prediction is a regression problem. In this paper the prediction task needs information from the past to perform the prediction. The state of art shows that the LSTM (Sepp Hochreiter, 1997) is one of the strongest model for this kind of problem.

### 3.1 Intention Prediction

This work focus on predicting driver's intention mainly turn right, turn left, stop and go straight action. Figure 2 illustrate all possible actions in a four way intersection, we observe 8 kind of action depending to driver direction. The two remaining actions

which are stopping and going straight whom are not presented in this figure. Thus for this task we have in total 10 intentions to predict or recognize, which means we have 10 classes, let $C = \{C_j, 1 \leq j \leq 10\}$ be the set of classes. Those actions can be maneuvered at any other kind of roads ( 2 ways intersection, 3 ways intersections, etc ...). Some of this actions can be forbidden according to the type of the road and the trafic regulation rules.

Datasets corresponding to each action were collected during a given interval of time, this interval starts few seconds before the action occurs and it ends few seconds after it occurs. The datasets collected before the actions occurrence were used for different time window size, to predict drivers's intentions.

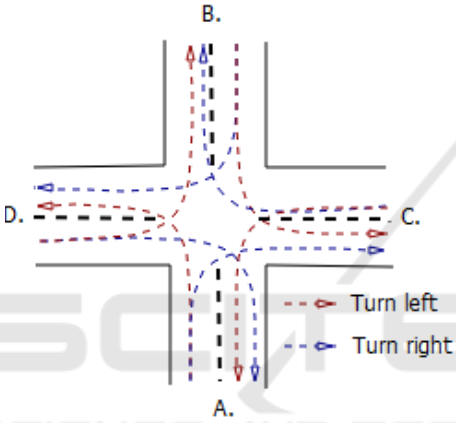


Figure 2: Four ways intersection.

### 3.1.1 Features

In addition to classical features (position, acceleration, velocity, etc...) we've defined an important feature that are effective for the prediction task. This feature made the model perform better, specially the recognition, we named it the directional tilt angle of the vehicle, we denote $\theta_t$ this angle. This angle is created between the abscissa axis and the vector created by the past position $(x_{t-1}, y_{t-1})$ and the present position $(x_t, y_t)$ of the vehicle.

The angle $\theta_t$ varies according to the vehicle direction, in other words it depends on the positions $(x_{t-1}, y_{t-1})$ and $(x_t, y_t)$ of the vehicle. Figure 3 illustrates how the angle is created for different cases, the angle $\theta_t$ is given by the fellowing equations:

- If $x_t < x_{t-1}$ and $y_t \geq y_{t-1}$, this corresponds to the case a) of figure 3, where $\theta_t$ is given by:

$$\theta_t = 180 - \arccos\left(\frac{|x_t - x_{t-1}|}{\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}}\right) \quad (7)$$

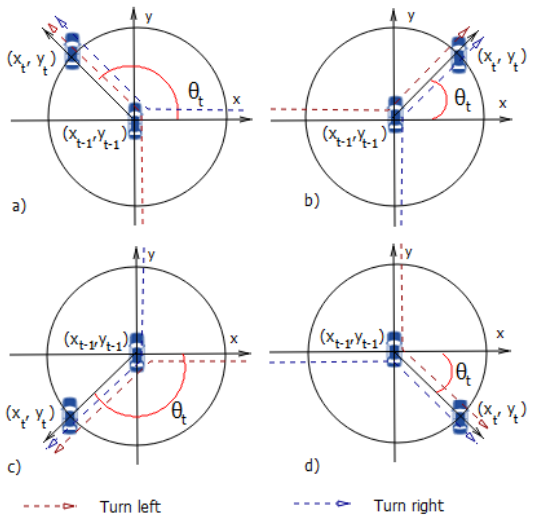


Figure 3: The directional tilt angle of the vehicle.

- If $x_t > x_{t-1}$ and $y_t \geq y_{t-1}$, this corresponds to the case b) of figure 3, where $\theta_t$ is given by :

$$\theta_t = \arccos\left(\frac{|x_t - x_{t-1}|}{\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}}\right) \quad (8)$$

- If $x_t \leq x_{t-1}$ and $y_t < y_{t-1}$, this corresponds to the case c) of figure 3, where $\theta_t$ is given by :

$$\theta_t = \arccos\left(\frac{|x_t - x_{t-1}|}{\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}}\right) - 180 \quad (9)$$

- If $x_t \geq x_{t-1}$ and $y_t < y_{t-1}$, this corresponds to the case d) of figure 3, where $\theta_t$ is given by :

$$\theta_t = -\arccos\left(\frac{|x_t - x_{t-1}|}{\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}}\right) \quad (10)$$

For the remaining case where, $x_t = x_{t-1}$ and $y_t = y_{t-1}$, which correspond to stopping action, $x_t - x_{t-1} = y_t - y_{t-1} = 0$, in this case $\theta_t$ can't be computed using the arccos function. For this sake $\theta_t$ is set to 1. For going straight intention $\theta_t$ can take the value 0, $\frac{\pi}{2}$, $-\frac{\pi}{2}$, or $\pi$ depending on the direction of the vehicle.

For the remaining intention maneuvering the angle $\theta_t$ takes value in the interval $]-\pi, \pi]$. Let's take the case a) of figure 3, if we refer to the figure 2 we can observe that we have two possible intentions beside going straight an stopping intentions: the intention of going from A to D and the intention of going from C to B. The first intention corresponds to a turn left for this case the angle $\theta_t$ varies from $\frac{\pi}{2}$ to $\pi$, while the second intention corresponds to a turn right, where the angle $\theta_t$ varies from $\pi$ to $\frac{\pi}{2}$.

#### 3.1.2 The Model

The model used in this paper is a four layers LSTM with tree hidden layers of 128 neurones each and an output layer with 10 neurones corresponding each to a given class $C_j$, $1 \leq j \leq 10$. The hidden layers use the rectified linear unit (ReLU) as activation function. The output layer use the softmax function to compute the probability that the observation is the in a given class. This model use as loss function the categorical crossentropy given by the fellowing equation:

$$L(Y,\hat{Y}) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{N} 1_{o_i \in C_j} \log(P(o_i \in C_j)) \quad (11)$$

Where, $N$ is the number of classes, $M$ the number of observation, $o_i$ the $ith$ observation, $Y$ vector of the truth labels, and $\hat{Y}$ the vector of predicted labels.

To find the minimum of the loss function or approximate it the model uses a stochastic optimization method called the Adam optimized (D. P. Kingma, 2015). The model is trained on dataset for different time windows size and different number of features in order to select the those that gives the best performances. The model was tested on data coming from time windows situated at different time to intention occurrence values, to evaluate its performances in term of it.

### 3.2 Trajectory Prediction

The trajectory prediction method use the same features as the intention prediction one. Since the trajectory prediction is mainly about predicting a sequence of the vehicle's positions in future, we denote $\hat{T}$ the predicted trajectory, $\hat{T} = \{(\hat{x}_k, \hat{y}_k) \in \mathbb{R}^2 : m+1 \leq k \leq K+m\}$, where $K$ is the number of the predicted positions, and $m$ is the length of the previous time window used to prediction $(x_k, y_k)$, let $L$ be the size of the window in second $L = \frac{m}{f_s}$, where $f_s = 10$ Hz is the frequency of data collection.

Since $(x_k, y_k)$ are in $\mathbb{R}^2$, this means we are facing a regression task. To predict trajectories we use the same model used for intention prediction with different loss function and different output layer. The loss function in this case is the mean squared error. The model was trained on different time window size for different number of features in order to select those which give better performances.

#### 3.2.1 Prediction of One Point of the Trajectory

To predict the point $(x_t, y_t)$ of the trajectory, the model takes as input, the previous $m$ sequence of features collected during the past time window of size $L$. Let $F_t = [f_{1,t}, ..., f_{n,t}]$ be the sequence of features at time $t$ and $n$ the number of features, the model uses the previous $m$ sequences of features to predict the vector $F_t$. The first two features of each vector of features corresponds to the position of the vehicle. In this part we are just interested in predicting only the next position of the vehicle, thus there is no need to predict all features.

#### 3.2.2 Prediction of a Sequence of Points

To predict a sequence of positions $T$ ( trajectory to be predicted), we first start by predicting the first next point $(x_t, y_t)$ of the trajectory by predicting $F_t$. As in the previous case the model takes the previous $m$ sequence of the feature as input to perform the prediction. To predict the position $(x_t, y_t)$ of a vehicle, the model takes a sequences of features as input, for this reason and in order to be able to predict the following position $(x_{t+1}, y_{t+1})$ of the vehicle, the model predict the sequence $F_t$ of features.

The predicted sequence of features $F_t$ and the last $m-1$ sequences of features are given to the model as input to predict the following sequence $F_{t+1}$ which contains the point $(x_{t+1}, y_{t+1})$ of the trajectory $\hat{T}$, this step is repeated till we predict all points of the trajectory. The disadvantage of this technic is that the error made on predicting $F_t$ will affect the prediction of the next sequence of features $F_{t+1}$ thus it will affect the prediction of $(x_{t+1}, y_{t+1})$.

## 4 EXPERIMENT

In this section we use the 3/4 of the database to train the model and the remaining data is used as validation set. Datasets with different time windows size were constituted. In this part we compare the model performances by varying time window size and the number of features used to train the model, for both trajectory prediction and intention prediction. For intention prediction we study the model performances according to time to intention occurrence.

### 4.1 Data Collection

Our model use data collected from the driving simulator CARLA (A. Dosovitskiy and V. Koltun, 2017). CARLA is an open source software developed by Alexey Dosovitski and al. at the computer and vision center of Barcelona. The CARLA simulator use a virtual environment which represents maps of virtual towns. These maps use a cartesian coordinate

system which allows us to locate vehicles and get its positions in time. Data collection code have been run on the town 01 of CARLA which contains two, three and four ways intersections, red lights and some trafic signs.

Several vehicles have been spawned on the town 01 of CARLA, with an autopilots which generates the trajectory of each vehicle and follow it. Vehicle's data is collected for a constant time step of 0.1 $s$ in other words data is collected at a frequency of $f_s$=10 Hz. Only data corresponding to intentions that we are interested in is stored.

## 4.2 Model Selection and Feature Selection

To evaluate the model performances, several comparisons were done on the model trained on different time window size and different number of features to select those that give the best performances.

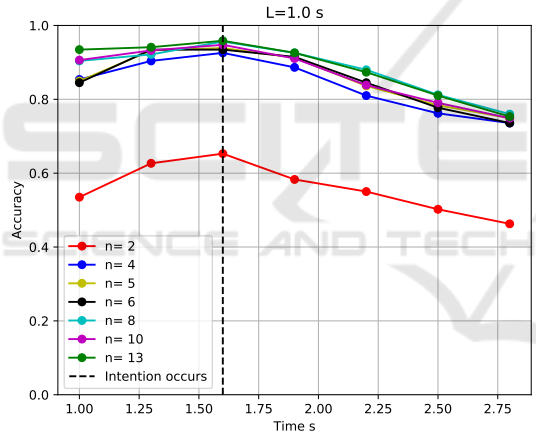


Figure 4: Accuracy of the model trained on time window of 1 $s$ vs time, for different number of features $n$.

Figure 4 illustrates the accuracy of the model for different time to intention occurrence, for a time window of size $L$=1 $s$. We can observe that the model trained on $n = 5$ features which are the positions, the velocity and the directional tilt angle of the vehicle, and the model trained on $n = 13$ features, which contains some road characteristics like trafic light and the type of the road at a given distance, have almost the same the performances. The accuracy of each model depends on the time to intention occurrence and the number of features. We can observe on this figure that if the model is trained just on the vehicle positions as features ($n = 2$) its accuracy become weak, while when the model is trained on more features its performances become better.

The accuracy of intention prediction increase when time to intention occurrence decreases despite the number of features. After the intention occurrences and the end of its maneuvering, the accuracy decrease which is normal because we didn't give the model the following intentions labels.

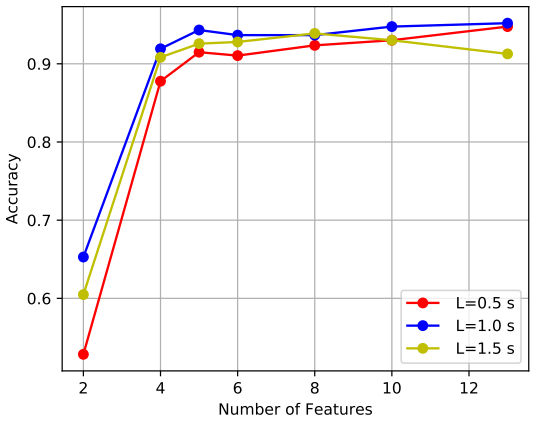


Figure 5: Intention prediction: Model accuracy vs the number of feature.

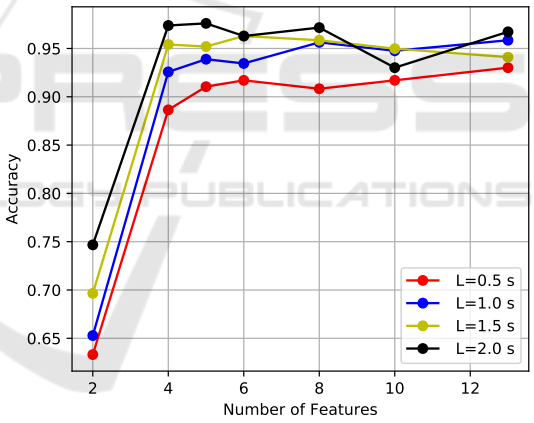


Figure 6: Intention recognition: Model accuraracy vs the number of feature.

Figure 5 and figure 6 show respectively the accuracy of the model for intention prediction and intention recognition, where the model was trained on different time window size for different number of features. We can observe that intention recognition performs better with long time window, where the accuracy can reach 97% for a time window of 2 $s$, while for intention prediction the accuracy reach 96% for a time window of 1 $s$.

Figure 7 illustrates the accuracy of the trajectory prediction model for one point prediction, for different time window size. Curves show that the model performs better with small set of features and long

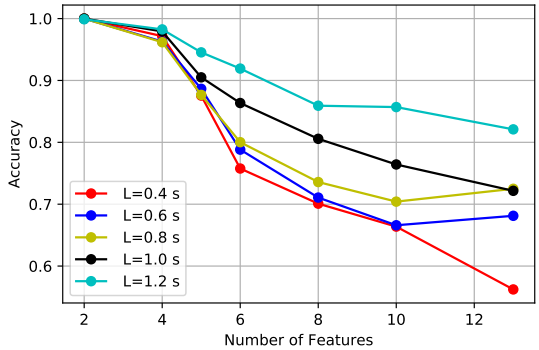


Figure 7: Prediction of one point of the future trajectory: model accuracy vs the number of feature, for different time window size $L$..

time window. We have an accuracy of 99.9% for the model trained only on vehicles positions (number of features $n = 2$), when the number of features increases the accuracy of the model decreases.

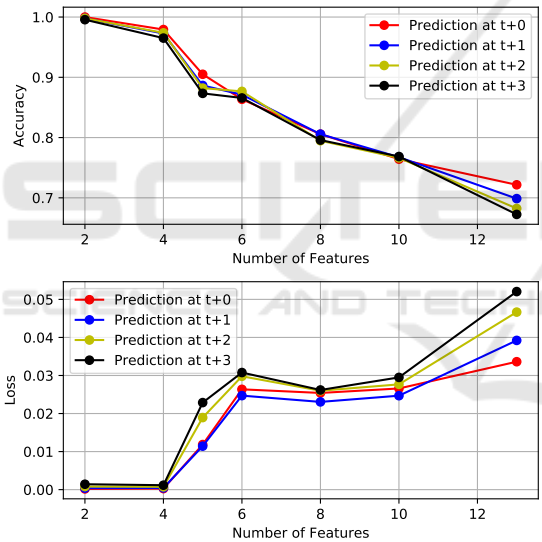


Figure 8: Trajectory prediction: model accuracy and loss vs $n$ the number of features , for a time window of $L$=1 $s$..

Figure 8 present the accuracy of the model, where we predict a sequence $(x_t, y_t), (x_{t+1}, y_{t+1}),...$ of future positions that constitute the predicted trajectory $\hat{T}$. By observing the figure 8, we can conclude that the trajectory prediction accuracy decreases when the number of features increases which is normal due to errors made on features prediction. When the number $n$ of features is high, it become difficult to predict all of those features without making errors.

Figure 9 illustrate the accuracy of the model at each step (prediction of a sequence of features) of the trajectory prediction, which is the accuracy variation according to time of prediction. We can observe that

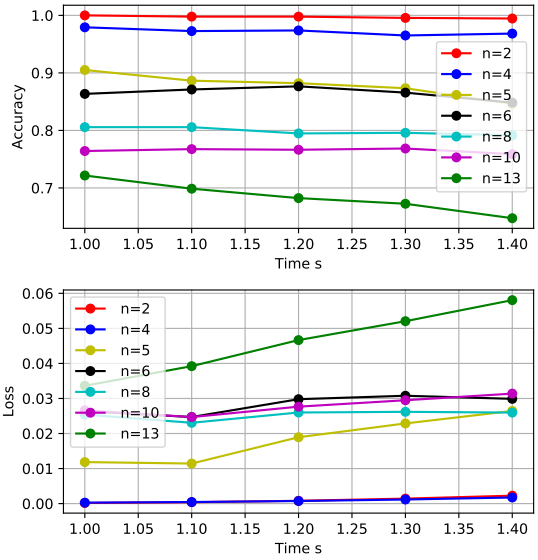


Figure 9: Trajectory prediction: model accuracy and loss vs time of prediction, for a time window of $L$=1 $s$..

the error made on predicting the sequence $F_t$ at each time $t$ in the process of trajectory prediction doesn't affect that much the performances of the prediction model. The accuracy is almost stable for low number of features while for high number of features the accuracy decreases slowly with time.

## 5 CONCLUSIONS

In this paper, we propose a method for driver intention prediction and recognition, and a method for trajectory prediction. With only a small set of features (4 or 5 features), this method gives high accuracy hence it performs in real time unlike the methods proposed in state of art, where a plenty of complex features are used to get good performances, which makes them greedy in term of computational time. This method will prevent collisions occurrence, whether it is used by a self-driving vehicle system or an ADAS systems. Trajectory prediction can be used for intention prediction where the predicted trajectory and features will be given as input to the intention prediction model to perform the intention prediction. We have shown in this paper that modeling this problem of prediction leads us to select the right features, which increases our model performances. Our feature selection method makes the model perform better with small dataset. By introducing the directional tilt angle of the vehicle as a feature our model performances increases. The proposed method gives an accuracy of 97% for intention recognition and an accuracy of 96% on intention prediction, whereas other work gets in

general 95% or less. The trajectory prediction model gives an accuracy of 99.9% just by using vehicle's positions and it gets 98% when the number of features is equal to 4. Thus for trajectory prediction when the number of predicted features increases the accuracy of the model decreases. This work will be extended to predict intentions at a roundabout and to predict the lane change intention, where the database will be updated with data coming from driver behavior for these intentions maneuvering.

## REFERENCES


A. Dosovitskiy, G. Ros, F. C. A. L. and V. Koltun, C. (2017). Carla: An open urban driving simulator. *1st Conference on Robot Learning*.

A. Zyner, S. W. and Nebot, E. (2018). Naturalistic driver intention and path prediction using recurrent neural networks. *IEEE Transactions on Intelligent Transportation Systems*.

B. Tang, S. Khokhar, R. G. (2015). Turn prediction at generalized intersections. *IEEE Intelligent Vehicles Symposium (IV)*.

D. P. Kingma, J. Lei Ba, A. (2015). A tutorial on mpeg/audio compression. *ICLR*.

Derek J. Phillips, T. A. W. and Kochenderfe, M. J. (2017). A tutorial on mpeg/audio compression. *IEEE Intelligent Vehicles Symposium (IV)*.

Djuric, N. and al. (2019). Multimodal trajectory predictions for autonomous driving using deep convolutional networks. *IEEE International Conference on Robotics and Automation (ICRA)*.

Hao Xue, D. Q. H. and Reynolds, M. (2018). Ss-lstm:a hierarchical lstm model for pedestrian trajectory prediction. *IEEE Winter Conference on Applications of Computer Vision*.

Hou, H. and al. (2011). Driver intention recognition method using continuous hidden markov model. *International Journal of Computational Intelligence Systems*, 4.

Laugier, C. and al. (2011). Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety. *IEEE Intelligent Transportation Systems*, 3:4–19.

L.R Rabiner, B. J. (1986). An introduction to hidden markov models. *IEEE ASSP Magazine*, 3:4–16.

Sepp Hochreiter, J. S. (1997). Long short-term memory, neural computation. *IEEE Multimedia*, 9:1735–1780.

Streubel, T. and Hoffmann, K. H. (2014). Prediction of driver intended path at intersections. *IEEE Intelligent Vehicles Symposium (IV)*.

Zhang, Y. and al. (2018). Integrating kinematics and environment context into deep inverse reinforcement learning for predicting off-road vehicle trajectories. *2nd Conference on Robot Learning*.