# Learning Vehicle Cooperative Lane-changing Behavior from Observed Trajectories in the NGSIM Dataset

Shuang Su, Katharina Muelling, John Dolan, Praveen Palanisamy, Priyantha Mudalige

***Abstract*—Lane-changing intention prediction has long been a hot topic in autonomous driving scenarios. However, none of the existing literature has taken both the vehicle's trajectory history and neighbor information into consideration when making the predictions. We propose a socially-aware LSTM algorithm in real world scenarios to solve this intention prediction problem, taking advantage of both vehicle past trajectories and their neighbor's current states. Simulation results show that these two components can lead not only to higher accuracy, but also to lower lane-changing prediction time, which plays an important role in potentially improving the autonomous vehicle's overall performance.**

***Keywords*—LSTM, social, lane-change intention.**

## I. Introduction

Lane changing has been regarded as one of the major factors causing traffic accidents [1]. As autonomous vehicles drive on highways, it is necessary for them to predict other vehicles' lane-changing intention to prevent potential collisions. There has been a lot of work attempting to model drivers' lane-changing behaviors, which can be divided into two types: rule-based algorithms and machine-learning-based algorithms.

Rule-based algorithms propose a set of rules to model lane changing. The most representative one is the 'gap acceptance model' [2], which assumed that drivers' lane-changing maneuvers are based on the lead and lag gaps in the target lane. The driver tends to make a lane change if the gap has attained a minimum acceptable value. Although straightforward and robust in simple scenarios, such methods need lots of fine-tuning for the corresponding parameters, which is tedious and time-consuming when many scenarios must be considered.

Machine-learning-based algorithms create a math model for the problem: given all of the vehicle-related features as input, and the vehicle's lane-changing intention as output, the methods try to optimize a classification model to obtain the best prediction results. A large number of classifiers such as logistic regression [3], SVM classifiers [4], and Bayesian network [5] have been adopted to formulate this model. However, none of them has considered both the vehicle's history trajectories and its neighbor information as features, which is partly due to the limitation of network structures not allowing those history or neighbor features.

A human-driven vehicle's lane-changing intention can be based on various factors, including the vehicle's own properties such as heading angle and acceleration, as well as its relationship to neighboring vehicles, such as its distance from the front vehicle. Recently, several references have explored the impact of neighboring traffic on the ego vehicle. Sadigh et al. proposed an algorithm for the ego vehicle to actively gather the surrounding cars' internal information through various sound-out actions [6]. Although seemingly attractive, such a strategy could only be adopted in toy scenarios. It would be unacceptable, for example, for an autonomous car to aggressively cut into another lane or wait for ages on highways just to test other vehicles' reactions. Recently, Alahi et al. came up with the idea of a social tensor, which encodes the neighbor agents' past trajectory information when predicting ego agent's future positions [7]. We use this idea, and bring in the social tensor from surrounding vehicles when predicting the ego vehicle's lane-changing intention.

Long short-term memory is a recurrent network structure which adopts 'forget gates' to prevent back-propagated errors from either exploding or vanishing [8]. It can therefore have a wide window slot and learn from events that happened hundreds of steps ago. Due to its ability to handle noisy, incompressible input sequences, it has been widely applied in numerous fields including speech processing [9], music composition [10], handwriting recognition [11], and even robot behavior planning [12]. We take advantage of this ability to introduce vehicle history trajectory information into the lane-change intention prediction process.

To get a full understanding of naturalistic driving behaviors, a large number of driving and lane-changing trajectories are required for the learning process, which is why we chose the NGSIM data set [13] to train and verify our algorithms. We also adopted a julia-based NGSIM [14] platform to extract input features for the network and visualize the traffic scenarios.

S. Su is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213 USA, E-mail: shuangs@andrew.cmu.edu.

K. Muelling and J. Dolan are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213 USA, E-mail: kmuelling@nrec.ri.cmu.edu, jmd@cs.cmu.edu.

P. Palanisamy and P. Mudalige are with Research and Development, General Motors Company. E-mail: praveen.palanisamy@gm.com

This paper is a preprint (IEEE "accepted" status). IEEE copyright notice.©2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. DOI:

Section II describes the procedures of extracting and pre-processing data from the NGSIM data set. In section III we introduce the input features and explain the specific network structure in detail. We then illustrate and compare our results with other methods' outcomes in section IV. Section V then summarizes the basic contents of our work and proposes potential future work.

## II. Data extraction and process

The open source Federal Highway Administration's Next Generation Simulation (NGSIM) data set [13], which has been adopted in numerous previous studies [3], [4], [15], [6], was picked to extract vehicle trajectories and build the lane-changing prediction model. At 0.1-second intervals, the data set recorded the location, speed, acceleration, and headway information for each vehicle on U.S. Highway 101 [16] and the Interstate 80 (I-80) Freeway [17]. Both locations contain 45 min. of vehicle trajectory data. Highway 101 is 640m long with five main lanes and a sixth auxiliary lane, while I-80 is about 500m in length, with six main lanes.

We extracted six sequences of vehicle trajectory data, 10 minutes each, from NGSIM. For each sequence, the first 2 minutes were defined as the test set, and the remaining 8 minutes as the training set. Since the data were recorded at 10 frames per second, we could obtain 1200 test time steps and 4800 training time steps in total.

A vehicle is labeled as 'intend to change lane to the left', 'intend to do car following', or 'intend to change lane to the right' at each time step. The way we labeled the vehicle status is as follows.

As depicted in Fig. 1, we first gathered all of the lane-changing points, i.e., the points where the vehicle crossed the dashed line dividing the lanes, for each vehicle. If a vehicle was on a lane-changing point at time step t, we checked its trajectories in [t-$\delta$t, t+$\delta$t] ($\delta$t=2s), and calculated its heading orientation $\theta$ during that time period. We then marked the starting point and ending point of this lane-changing trajectory when $\theta$ has reached a bounding value $\theta_{bound}$: $|\theta| = \theta_{bound}$.

Fig. 2 described the way we collected the trajectory pieces. $n$ consecutive time steps were packed into one trajectory piece for each vehicle. If the $n$th time step of a trajectory piece was a lane-changing time step, then the piece was a lane-changing piece, otherwise it was labeled as a car-following piece. The trajectory pieces were collected in a 'shifting' manner to make the most use of the data. In this paper, we set $n$ to be 6, 9, and 12 to determine the impact of length of the history trajectories on the final results.

We could then get around 60,000 lane changing pieces plus 400,0000 car following pieces in total for training. This clearly involves a data-imbalance problem, where there are far more car following pieces than lane-changing pieces for training, which will result in over-fitting in the training process. To deal with this problem, we randomly selected the same number of pieces, $N$, from the lane-changing-left pool, the car-following pool and the lane-changing-right pool, mixing them together for the training data set. To make the most use of the data, $N$ is set to be the number of pieces in the lane-changing-right pool.

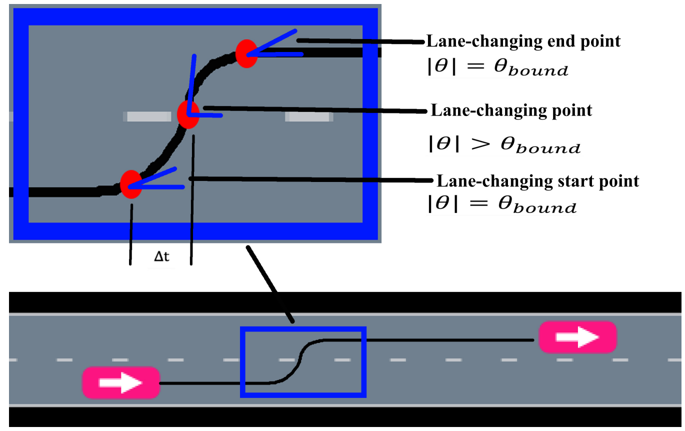



Fig. 1. The start point, lane-changing point, and end point of a lane-changing trajectory.

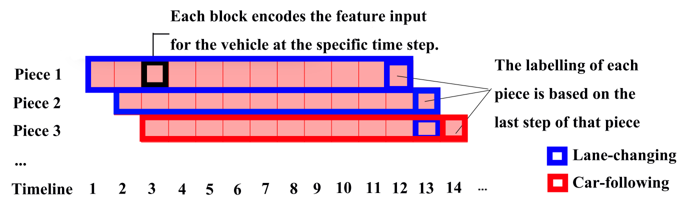



Fig. 2. Shifting methods of extracting input features and output lane-changing intention for one vehicle. n continuous time steps were packed into one trajectory piece. If the $n$th time step of a trajectory piece was a lane-changing time step, then the piece was a lane-changing piece (as depicted in Piece 1 and Piece 2, which was marked as blue), otherwise it was labeled as a car-following piece (as depicted in Piece 3, which was marked as pink). The first time step of the collected pieces shifted one step at a time so that we could make the most use of the data.

Each vehicle's lane-changing intention was then predicted at each time step given its previous 11-time-step history trajectories and neighbor information in the test set. The lane-changing prediction time was also calculated after filtering the results. Specifically, a lane-changing prediction point is settled if a vehicle is predicted to make a lane change for 3 continuous time steps, and the lane-changing prediction time is defined to be the time gap between the lane-changing point and the lane-changing prediction point, as depicted in Fig. 3.

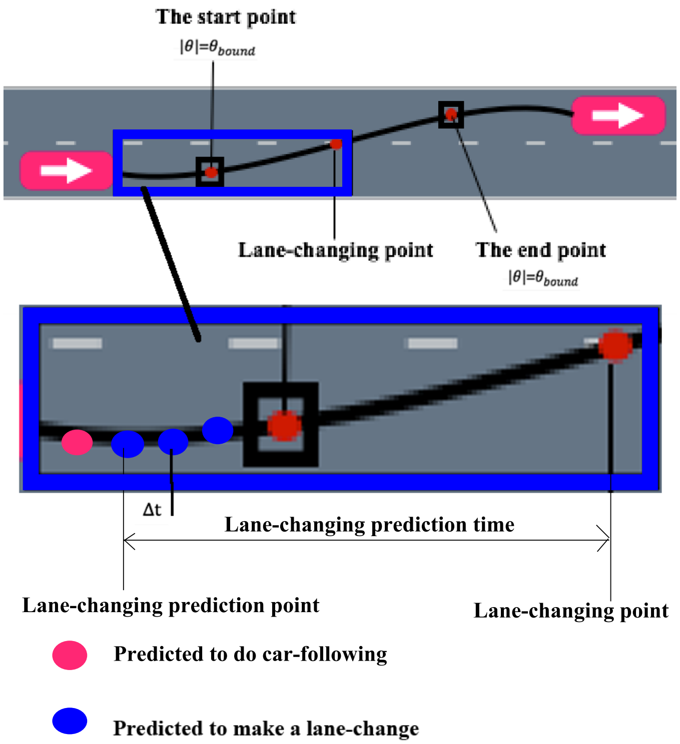



Fig. 3. A lane-changing prediction point is settled if a vehicle is predicted to make a lane change for 3 continuous time steps. The lane-changing prediction time is defined to be the time gap between the lane-changing point and the lane-changing prediction point.

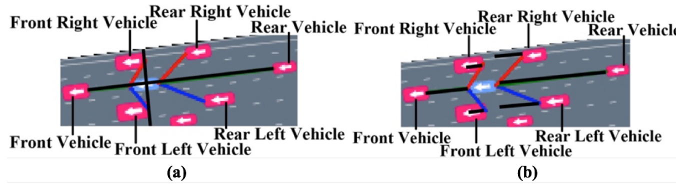



Fig. 4. Neighbor information collection. In (a), we first divided the neighbor space into four parts based on the ego vehicle's orientation and center position, and defined the corresponding neighbor vehicles based on their relative positions to the ego vehicle. We then collected the longitudinal distances between these neighbor vehicles and the ego vehicle to be the neighbor features in (b).

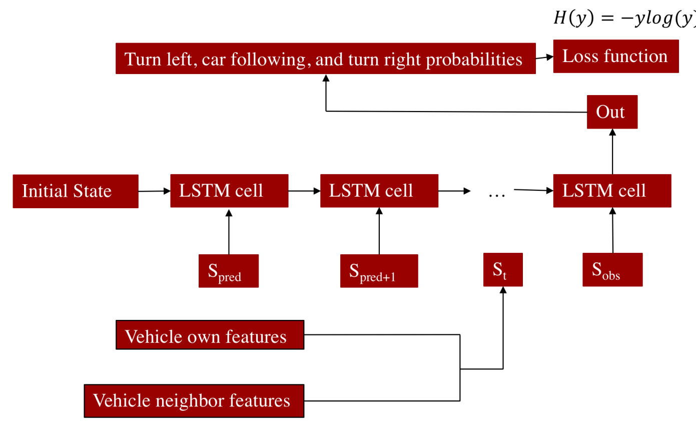



Fig. 5. LSTM network structure for lane-changing intention prediction.

## III. Methodology

### A. Input features

The input features for each vehicle at each time step are:
a) the vehicle's own information
1. vehicle acceleration
2. vehicle steering angle with respect to the road
3. the global lateral vehicle position with respect to the lane
4. the global longitudinal vehicle position with respect to the lane

b) the vehicle's neighbor information (see Fig. 4, "ego vehicle" here refers to the vehicle whose lane-changing intention we are estimating)
1. the existence of left lane(1 if existed, 0 if not)
2. the existence of right lane(1 if existed, 0 if not)
3. the longitudinal distance between ego vehicle and left-front vehicle
4. the longitudinal distance between ego vehicle and front vehicle
5. the longitudinal distance between ego vehicle and right-front vehicle
6. the longitudinal distance between ego vehicle and left-rear vehicle
7. the longitudinal distance between ego vehicle and rear vehicle
8. the longitudinal distance between ego vehicle and right-rear vehicle

### B. Network structure

We adopted the LSTM network structure, as depicted in Fig. 5, to deal with this problem. The embedding dimension chosen for the vehicle's own features as well as its neighbor features was 64, and the hidden dimension for the LSTM network was 128. We selected the learning rate to be 0.000125, using soft-max cross entropy loss as the training loss.

## IV. Results

### A. Comparison with other network structures

We obtained and compared our results with feedforward neural network, logistic regression and LSTM without neighbor feature inputs to show the advantages of adding history trajectories and social factors.

Table I and Fig. 6 show the classification accuracy rate calculated by our algorithm, feedforward neural network, and logistic regression. Social-LSTM, based on the benefits of history trajectory information, outperforms the other two methods in all classification types (lane-changing left, car-following, and lane-changing right) in terms of prediction

| | Real Predict | Left | Following | Right |
|---|---|---|---|---|
| Social-LSTM | Left | 87.40% | 12.34% | 0.26% |
| | Following | 7.47% | 85.33% | 7.20% |
| | Right | 2.94% | 11.22% | 85.84% |
| Feedforward Neural Network | Left | 84.6% | 15.40% | 0% |
| | Following | 2.61% | 83.78% | 13.61% |
| | Right | 2.44% | 12.91% | 79.65% |
| Logistic Regression | Left | 64.91% | 35.03% | 0.06% |
| | Following | 9.88% | 82.87% | 7.25% |
| | Right | 0.05% | 36.30% | 63.65% |

TABLE I. LANE-CHANGING PREDICTION ACCURACY COMPARISON AMONG SOCIAL-LSTM, FEEDFORWARD NEURAL NETWORK, AND LOGISTIC REGRESSION.

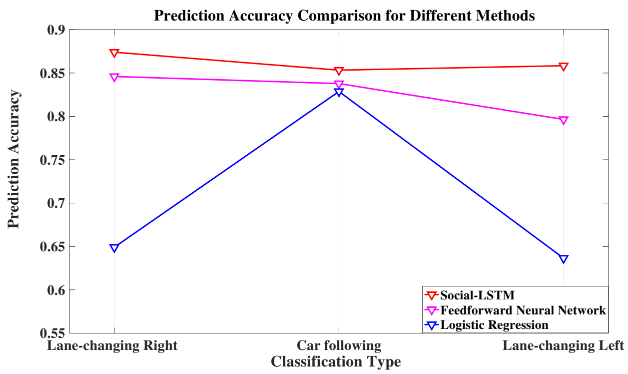



Fig. 6. Prediction accuracy comparison for different methods. Social-LSTM outperforms the other two in all classification types including lane-changing right, car-following, and lane-changing right.

accuracy.

## B. Comparison among different trajectory lengths

We then obtained and compared the predicted accuracy rate for different trajectory lengths. Specifically, we set the history trajectory length of the LSTM structure to be 6, 9, and 12, comparing them against each other. The results are shown in Table II, and visualized in Fig. 7. The prediction accuracy increases as the history length grows in all prediction scenarios (lane-changing left, car following, and lane-changing right).

## C. Comparison between with- and without- neighbor scenarios

Table III depicts the lane-changing prediction time generated by with- and without- neighbor-input-feature LSTM models, which is defined to be the time gap between the time at which the model predicts there will be a lane-change and the time at which the vehicle has actually reached the lane-changing point. The longer the time gap is, the more useful the prediction is. It can be seen from Fig. 8 that adding neighbor features prolongs the lane-changing prediction time in most cases.

| Social-LSTM, History Length=12 | | | | |
|---|---|---|---|---|
| | Real Predict | Left | Following | Right |
| Sequence 1 | Left | 87.69% | 11.72% | 0.60% |
| | Following | 11.80% | 84.55% | 3.65% |
| | Right | 0.00% | 12.66% | 87.34% |
| Sequence 2 | Left | 84.85% | 15.00% | 0.15% |
| | Following | 7.49% | 88.71% | 3.79% |
| | Right | 4.79% | 7.56% | 87.66% |
| Sequence 3 | Left | 98.17% | 1.83% | 0.00% |
| | Following | 14.32% | 81.42% | 4.26% |
| | Right | 13.04% | 0.00% | 86.96% |
| Sequence 4 | Left | 90.76% | 8.61% | 0.63% |
| | Following | 3.94% | 81.59% | 14.46% |
| | Right | 0.58% | 12.14% | 87.28% |
| Sequence 5 | Left | 92.11% | 7.89% | 0% |
| | Following | 2.71% | 90.13% | 7.16% |
| | Right | 7.16% | 17.65% | 75.19% |

(a)

| Social-LSTM, History Length=9 | | | | |
|---|---|---|---|---|
| | Real Predict | Left | Following | Right |
| Sequence 1 | Left | 86.94% | 12.56% | 0.49% |
| | Following | 12.87% | 83.34% | 0.79% |
| | Right | 0.00% | 14.50% | 85.50% |
| Sequence 2 | Left | 83.13% | 16.80% | 0.07% |
| | Following | 7.94% | 88.01% | 4.04% |
| | Right | 4.65% | 7.82% | 87.53% |
| Sequence 3 | Left | 98.17% | 1.83% | 0.00% |
| | Following | 14.99% | 80.54% | 4.48% |
| | Right | 14.24% | 0.00% | 85.76% |
| Sequence 4 | Left | 89.82% | 9.69% | 0.49% |
| | Following | 4.41% | 80.04% | 15.55% |
| | Right | 2.69% | 11.78% | 85.53% |
| Sequence 5 | Left | 90.05% | 9.95% | 0.00% |
| | Following | 3.53% | 89.60% | 6.86% |
| | Right | 5.61% | 21.08% | 73.30% |

(b)

| Social-LSTM, History Length=6 | | | | |
|---|---|---|---|---|
| | Real Predict | Left | Following | Right |
| Sequence 1 | Left | 84.27% | 15.44% | 0.29% |
| | Following | 23.49% | 72.20% | 4.31% |
| | Right | 3.30% | 28.66% | 68.04% |
| Sequence 2 | Left | 77.21% | 20.82% | 1.98% |
| | Following | 16.80% | 78.72% | 4.48% |
| | Right | 8.55% | 13.06% | 78.38% |
| Sequence 3 | Left | 95.12% | 4.88% | 0.00% |
| | Following | 24.84% | 70.33% | 4.84% |
| | Right | 14.42% | 12.50% | 73.08% |
| Sequence 4 | Left | 78.47% | 20.21% | 1.32% |
| | Following | 8.90% | 72.40% | 18.70% |
| | Right | 2.15% | 16.76% | 81.09% |
| Sequence 5 | Left | 77.78% | 22.22% | 0.00% |
| | Following | 13.92% | 81.87% | 4.21% |
| | Right | 0.00% | 30.01% | 69.99% |

(c)

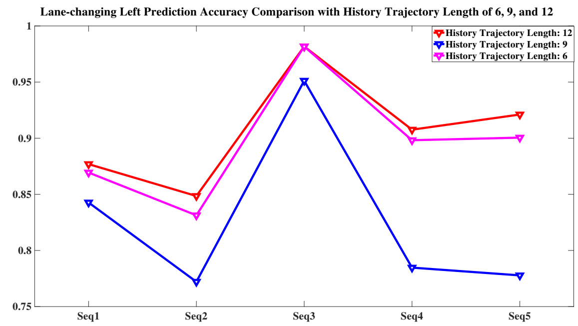
TABLE II. LANE-CHANGING PREDICTION ACCURACY COMPARISON AMONG DIFFERENT TRAJECTORY LENGTHS

| | Lane-changing Left | Lane-changing Right |
|---|---|---|
| Sequence 1 | 1.17s/1.08s | 1.00s/0.70s |
| Sequence 2 | 1.34s/1.31s | 1.39s/1.31s |
| Sequence 3 | 1.59s/1.58s | 1.48s/1.50s |
| Sequence 4 | 1.66s/1.50s | 1.32s/1.33s |
| Sequence 5 | 1.66s/1.55s | 0.75s/0.73s |
| Average | 1.44s/1.31s | 1.14s/1.03s |

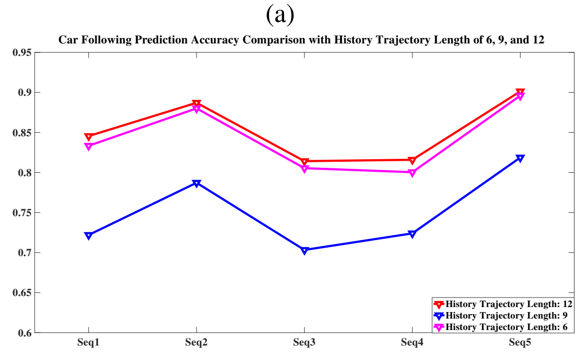
TABLE III. LANE-CHANGING PREDICTION TIME COMPARISON BETWEEN WITH-NEIGHBOR AND WITHOUT-NEIGHBOR SCENARIOS

# V. CONCLUSIONS

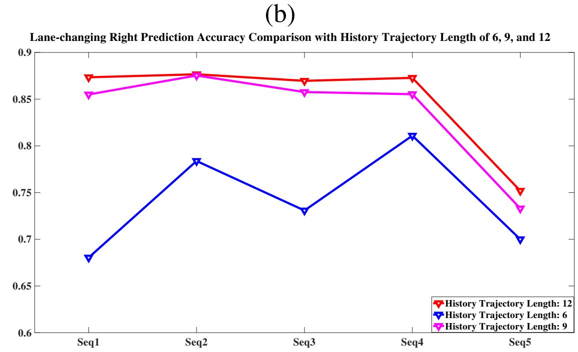
This paper proposes a LSTM network structure with the introduction of neighbor vehicles' features to make lane-changing intention predictions for each individual vehicle. We compared our methods with different network structures such




(a)



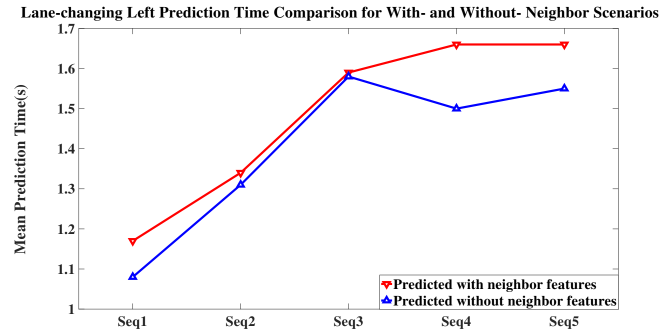

(b)



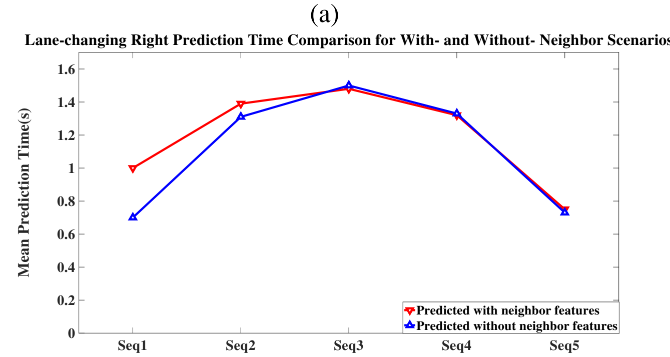

(c)

Fig. 7. We compared the prediction accuracy for different history trajectory lengths, the history time step length in the LSTM network structure. For each test sequence, the prediction accuracy increased as the history trajectory length grew.

as feed-forward neural network and logistic regression, as well as with LSTM structures without neighbor features to show the advantages of adding time and space information. We also compared the structure among different history trajectory lengths, and saved their influence on the final prediction results. Future work will mainly focus on extending the algorithm into practical scenarios, and seeing if the trained network can be adopted on a real autonomous-driving car. The outstanding performance of the LSTM network also suggests the potential of attempting other recurrent network structures to further improve the prediction results in traffic scenarios.




(a)




(b)

Fig. 8. Prediction time comparison for with- and without- neighbor scenarios. Both lane-changing left and lane-changing right predictions show a increase, if not maintained, in prediction time after adding neighbor features.

## ACKNOWLEDGMENTS

The authors would like to thank General Motors Company for their sponsorship and great help.

## REFERENCES

[1] D. Clarke, J. W. Patrick, and J. Jean, "Overtaking road-accidents: Differences in manoeuvre as a function of driver age," *Accident Analysis & Prevention*, vol. 30, no. 4, pp. 455–467, 1998.

[2] Q. Yang and N. K. Haris, "A microscopic traffic simulator for evaluation of dynamic traffic management systems," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 3, pp. 113–129, 1996.

[3] C. Oh, C. J, and P. S, "In-depth understanding of lane changing interactions for in-vehicle driving assistance systems," *International journal of automotive technology*, vol. 18, no. 2, pp. 357–363, 2017.

[4] Y. Dou, Y. Fengjun, and F. Daiwei, "Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers," in *IEEE International Conference on Advanced Intelligent Mechatronics*. IEEE, 2016, pp. 901–906.

[5] A. Kuefler, J. Morton, and T. Wheeler, "Imitating driver behavior with generative adversarial networks," in *IEEE Transactions on Intelligent Vehicles*. IEEE, 2017, pp. 204–211.

[6] D. Sadigh, S. Sastry, S. A. Seshia, and A. Dragan, "Planning for autonomous cars that leverage effects on human actions," in *Robotics: Science and Systems*. IEEE, 2016.

[7] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, F. Li, and S. Savarese, Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2016, pp. 961–971.

[8] S. Hochreiter and S. Jurgen, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] H. Sak, S. Andrew, and B. Francoise, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth Annual Conference of the International Speech Communication Association*. IEEE, 2014.

[10] D. Eck and S. Juergen, "A first look at music composition using lstm recurrent neural networks," in *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*. IEEE, 2002.

[11] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[12] Y. Huang and S. Yu, "Learning to pour," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2017.

[13] "Next generation simulation fact sheet," in *ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm*. Federal Highway Administration, 2011.

[14] T. Wheeler. A julia package for handling the next generation simulation (ngsim) traffic dataset. [Online]. Available: https://github.com/sisl/NGSIM.jl

[15] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, A. Wedel, and W. Rosenstiel, "Object-oriented bayesian networks for detection of lane change maneuvers," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, no. 3, pp. 19–31, 2012.

[16] J. Colyar and J. Halkias, "Us highway 101 dataset," *Federal Highway Administration(FHWA)*, vol. Tech, no. Rep, 2007.

[17] ——, "Us highway 80 dataset," *Federal Highway Administration(FHWA)*, vol. Tech, no. Rep, 2006.

View publication stats