# Final Presentation: Mid Semester Review

Ankit Khandelwal

Subject: An LSTM Network for highway trajectory prediction of a vehicle with given trajectories of the surrounding vehicles

Master's Degree Program in International Automotive Engineering (IAE)
Abteilung für Elektrotechnik und Informatik
Technische Hochschule Ingolstadt

12th December, 2019

Technische Hochschule Ingolstadt

# Outline

## An LSTM Network for Highway Trajectory Prediction

This paper present a first step towards consistent trajectory prediction by introducing a long short-term memory (LSTM) neural network, which is capable of accurately predicting future longitudinal and lateral trajectories for vehicles on highway.
Link: https://ieeexplore.ieee.org/abstract/document/8317913

# 1.2 Problem Statement

- We consider the problem of predicting future trajectories of vehicles driving on a highway, using previously observed data; these predictions can then be used to plan the motion of an autonomous vehicle.
- we consider a set of observable features I and a set of target outputs O to be predicted. We assume that the features can all be acquired simultaneously at regular intervals,We let $T = \{0, ..., K\}$ and for $x \in I, k \in T$.
- We denote by $x_k$ the value of feature x observed k time steps earlier. Similarly, we denote by $y_k$ the value of output $y \in O, k \in T$ time steps.

## 1.2 Problem Statement

- We use uppercase

$$X = (x_k), \, for \;\; x \in I, k \in T$$

and

$$Y = (y_k), \, for \;\; y \in O, k \in T$$

to respectively denote the tensors of the observed features and corresponding predicted outputs. We propose to use a machine learning approach, in which we train a regression function f such that the predicted outputs $\hat{Y} = f(X)$ match the actual values as closely as possible.

- Our approach is to train a predictor for the trajectory of a single "target" vehicle; in order to only use data which can realistically be gathered, we limit the amount of available information to the vehicles immediately around the target vehicle.

# 1.3 Data-set Used

- We use the Next Generation Simulation (NGSIM) dataset, collected in 2005 by the United States Federal Highway Administration, which is one of the largest publicly available source of naturalistic driving data and, as such, has been widely studied in the literature
- we consider the US101 dataset which contains 45 minutes of trajectories for vehicles on the US101 highway, between 7:50am and 8:35am during the transition from fluid traffic to saturation at rush hour.
- In total, the dataset contains trajectories for more than 6000 individual vehicles, recorded at 10 Hz. The NGSIM dataset provides vehicle trajectories in the form of $(X; Y)$ coordinates of the front center of the vehicle in a global frame, and of local $(x; y)$ coordinates of the same point on a road-aligned frame.

# 1.3 Data-set Used

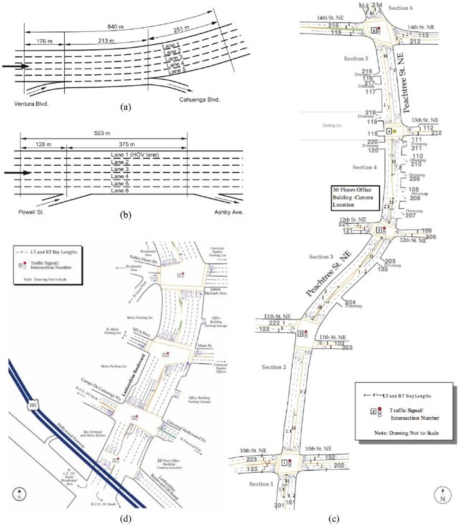



Fig1: Dataset Visualization

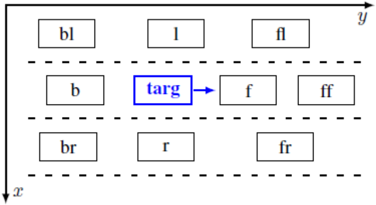



Fig2: Vehicle of interests around the Target Vehicle.

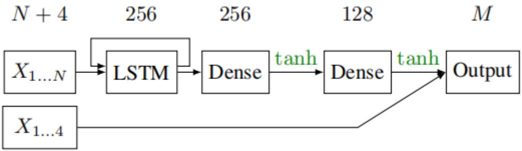



Fig3: Network Architecture.

# 1.5 Results by Author

TABLE I
RMS ERROR FOR THE TESTED MODELS

| Model | Prediction horizon 1 s | 2 s | 3 s | 4 s | 6 s | 8 s | 10 s |
|---|---|---|---|---|---|---|---|
| Reference* | **0.11** | **0.25** | **0.33** | **0.40** | 0.53 | 0.60 | 0.73 |
| Type* | 0.39 | 0.39 | 0.44 | 0.48 | 0.53 | 0.63 | 0.69 |
| No $ff$* | 0.14 | 0.24 | 0.33 | 0.41 | 0.54 | 0.65 | 0.76 |
| No bypass | 0.80 | 0.82 | 0.85 | 0.88 | 0.93 | 0.97 | 1.03 |
| Bypass before | 0.33 | 0.38 | 0.43 | 0.46 | 0.52 | 0.61 | 0.68 |
| Lin. activ. | 1.38 | 1.39 | 1.40 | 1.42 | 1.46 | 1.51 | 1.56 |
| 2 LSTMs | 1.25 | 1.26 | 1.28 | 1.29 | 1.33 | 1.37 | 1.41 |
| 3 dense* | 0.34 | 0.38 | 0.44 | 0.50 | 0.59 | 0.70 | 0.72 |
| [14] | **0.11** | 0.32 | 0.71 | not available | | | |
| Bagged | 0.17 | **0.25** | **0.33** | **0.40** | **0.46** | **0.57** | **0.65** |

(a) Lateral position (errors are in m)

| Model | Prediction horizon 1 s | 2 s | 3 s | 4 s | 6 s | 8 s | 10 s |
|---|---|---|---|---|---|---|---|
| Reference* | 0.71 | 0.99 | 1.25 | 1.49 | 2.10 | 2.60 | 2.96 |
| Type* | 0.65 | 0.88 | 1.05 | **1.25** | **1.75** | **2.28** | **2.74** |
| No $ff$* | 0.67 | 0.91 | 1.16 | 1.44 | 1.98 | 2.43 | 2.84 |
| No bypass | 1.50 | 1.50 | 1.55 | 1.66 | 2.05 | 2.50 | 2.89 |
| Bypass before | 0.78 | 0.90 | 1.06 | 1.26 | 1.76 | 2.30 | 2.78 |
| Lin. activ. | 0.77 | 1.10 | 1.34 | 1.56 | 2.08 | 2.58 | 2.94 |
| 2 LSTMs | 0.76 | 1.14 | 1.42 | 1.71 | 2.22 | 2.72 | 3.17 |
| 3 dense* | 0.73 | **0.87** | **1.04** | **1.25** | 1.76 | 2.30 | 2.77 |
| Bagged | **0.64** | **0.81** | **0.98** | **1.18** | **1.63** | **2.08** | **2.48** |

(b) Longitudinal speed (errors are in $\mathrm{m\,s^{-1}}$)

Fig4: Error Values.

# 1.5 Results

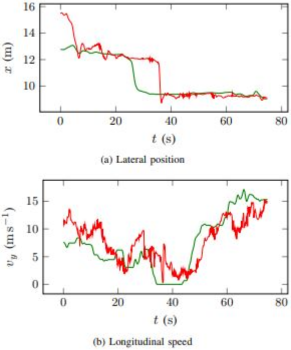



Fig5: Output: Represented in Normalized Values

# Outline

# 2.1 What is trajectory planning?

- Trajectory planning is moving from point A to point B while avoiding collisions over time. This can be computed in both discrete and continuous methods. Trajectory planning is a major area in robotics as it gives way to autonomous vehicles.
- Trajectory planning is sometimes referred to as motion planning and erroneously as path planning. Trajectory planning is distinct from path planning in that it is parametrized by time. Essentially trajectory planning encompasses path planning in addition to planning how to move based on velocity, time, and kinematics.

# 2.2 Types of trajectory prediction

There can be two possibilities:

- First is that the initial trajectory of the vehicle and surrounding is given and then predict the future motion.
- Second is that the trajectory of all surrounding vehicles are given and predict the motion of target, without any information given about target vehicle.(This one is used for this project)

# 2.3 What is time series data?

- A time series is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data.
- Because this is discrete data some sort of filtering is required to filter noise. In this application savitzky golay filter is used to remove noise from calculated difference values.

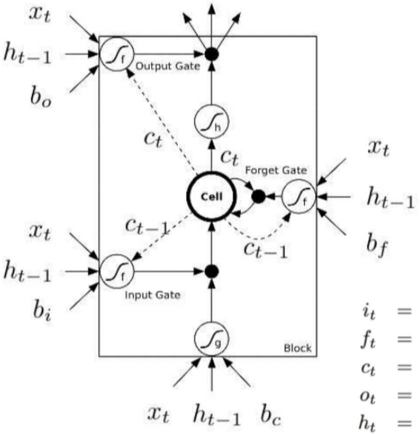
- One of the appeals of RNNs is the idea that they might be able to connect previous information to the present task, such as using previous video frames might inform the understanding of the present frame.
- Sometimes, we only need to look at recent information to perform the present task. RNNs can learn to use the past information.
- The LSTM network can perform this task of prediction of time series very efficiently because of its capability to forget the unused information in a long time series.

# Outline

# 3.1 What is LSTM network?

- Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies.
- LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn.

# 3.1 What is LSTM network?




$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)\\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)\\
c_t &= f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)\\
h_t &= o_t tanh(c_t)
\end{aligned}
$$

Fig6: Basic LSTM cell

# 3.2 Data-set pre-processing

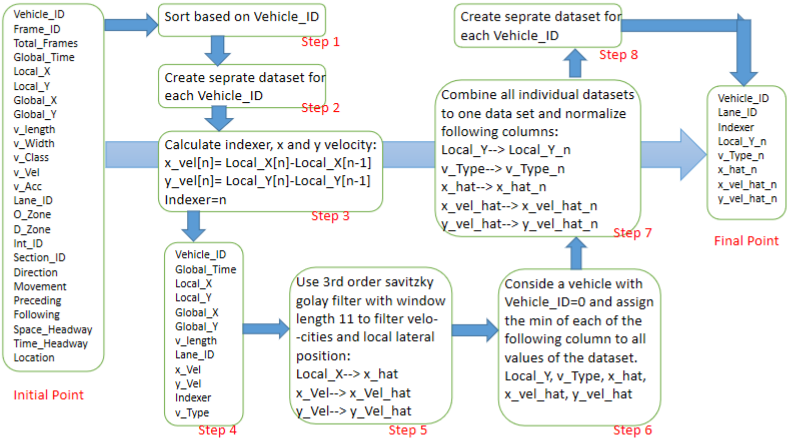



Fig7: Preprocessing Steps

# 3.2 Data-set pre-processing

- Step 0: First step is to figure out what are the important parameters required from initial data set to final data set.
- Step 1: Sort the dataset based on vehicle ids given.
- Step 2: Create separate csv file for each dataset with number of rows equal to minimum number of instances available for a vehicle out of all 600 vehicles. This step is done so that the feature vector has constant dimension. Also it will hep in velocity calculation in each direction
- Step 3: Calculate lateral x and longitudinal y direction velocities for all the vehicles and assign an indexer to each row because the time difference from the given global time is not possible to be calculated. Each index represents 100ms time.
- Step 4: Store the velocities and Index values in the dataset and drop the columns which are not required.

# 3.2 Data-set pre-processing

- Step 5: As the velocities are the first difference of the position values some noise will automatically gets included in the signal. To filter the velocities Savitzky-Golay filter is used, which is explained in more detail in Sub-Section **??**. Then the nun filetred values are replaced by the filtered values.
- Step 6: As the network will only take values between range {0,1}, the normalization of the values is required. To achieve this a new dataset is created manually with vehicle id 0..
- Step 7: Now the data of all the vehicle ids is concatenated to normalize the required columns. After normalization the non-normalized values are replaced by normalized values.
- Step 8: Sort the dataset again based on vehicle ids given as it was done in Step 1. Now we have the final datasets for all the vehicles which will be used to create input feature vector as shown in Figure 3.

# 3.3 Feature Vector creation

For the target vehicle, we define the following features:

- local lateral position $x_{targ}$, to account for different behaviors depending on the driving lane,
- local longitudinal position $y_{targ}$, to account for different behaviors when approaching the merging lane,
- lateral and longitudinal velocities $v_{x_{targ}}$ and $v_{y_{targ}}$,
- type (motorcycle, car or truck), encoded respectively as 1, 2 or 3.

For each vehicle $p \in \{bl; b; br; l; f; r; fl; fr; ff\}$, we define the following features:

- lateral velocity $v_{x_p}$,
- longitudinal velocity relative to targ: $v_{y_p} = v_{y_{targ}} - v_{y_p}$,
- lateral distance from targ: $x_p = x_p - x_{targ}$,
- longitudinal distance from targ: $y_p = y_p - y_{targ}$,
- type (motorcycle, car or truck), encoded respectively as 1, 2 or 3.

# 3.3 Feature Vector creation

As per the matrix represented in Fig2

Input Feature Vector:

| | |
|---|---|
| $X = (y_{targ}, v_{x_{targ}}, v_{type_{targ}},$ | Target Vehicle |
| $v_{x_{fl}}, \Delta v_{y_{fl}}, \Delta x_{fl}, \Delta y_{fl}, v_{type_{fl}},$ | Front Left Vehicle |
| $v_{x_{ff}}, \Delta v_{y_{ff}}, \Delta x_{ff}, \Delta y_{ff}, v_{type_{ff}},$ | Front Front Vehicle |
| $v_{x_{fr}}, \Delta v_{y_{fr}}, \Delta x_{fr}, \Delta y_{fr}, v_{type_{fr}},$ | Front Right Vehicle |
| $v_{x_{l}}, \Delta v_{y_{l}}, \Delta x_{l}, \Delta y_{l}, v_{type_{l}},$ | Left Vehicle |
| $v_{x_{f}}, \Delta v_{y_{f}}, \Delta x_{f}, \Delta y_{f}, v_{type_{f}},$ | Front Vehicle |
| $v_{x_{r}}, \Delta v_{y_{r}}, \Delta x_{r}, \Delta y_{r}, v_{type_{r}},$ | Right Vehicle |
| $v_{x_{bl}}, \Delta v_{y_{bl}}, \Delta x_{bl}, \Delta y_{bl}, v_{type_{bl}},$ | Back Left Vehicle |
| $v_{x_{b}}, \Delta v_{y_{b}}, \Delta x_{b}, \Delta y_{b}, v_{type_{b}},$ | Back Vehicle |
| $v_{x_{br}}, \Delta v_{y_{br}}, \Delta x_{br}, \Delta y_{br}, v_{type_{br}})$ | Back Right Vehicle |

Output Vector:

| | |
|---|---|
| $Y = (x_{targ}, v_{y_{targ}})$ | Target Vehicle |

# 3.4 Network Architecture

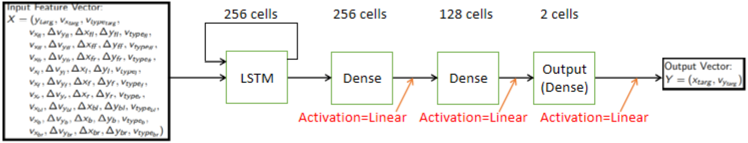



Fig8:Network Used

- Network is same except the first four instances are not directly given to output layer
- The activation are not tanh but linear in this network.
- Dense/fully connected layer: A linear operation on the layer's input vector.

# 3.4 Network Architecture

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 1500, 256)         312320
_________________________________________________________________
dense_1 (Dense)              (None, 1500, 256)         65792
_________________________________________________________________
dense_2 (Dense)              (None, 1500, 128)         32896
_________________________________________________________________
dense_3 (Dense)              (None, 1500, 2)           258
=================================================================
Total params: 411,266
Trainable params: 411,266
Non-trainable params: 0
_________________________________________________________________
None
```

Fig9:Network Definition

# 3.4 Network Architecture

TABLE I
HYPER PARAMETERS FOR MODEL LAYERS

| Layer Name | | lstm_1 | dense_1 | dense_2 | dense_3 |
|---|---|---|---|---|---|
| S.No. | Parameter | Values | Values | Values | Values |
| 1 | units | 256 | 256 | 128 | 2 |
| 2 | activation | - | linear | linear | linear |
| 3 | use_bias | - | True | True | True |
| 4 | kernel_initializer | - | 'glorot_uniform' | 'glorot_uniform' | 'glorot_uniform' |
| 5 | bias_initializer | - | 'zeros' | 'zeros' | 'zeros' |
| 6 | kernel_regularizer | - | None | None | None |
| 7 | bias_regularizer | - | None | None | None |
| 8 | activity_regularizer | - | None | None | None |
| 9 | kernel_constraint | - | None | None | None |
| 10 | bias_constraint | - | None | None | None |

Fig10:Hyper Parameters of layers

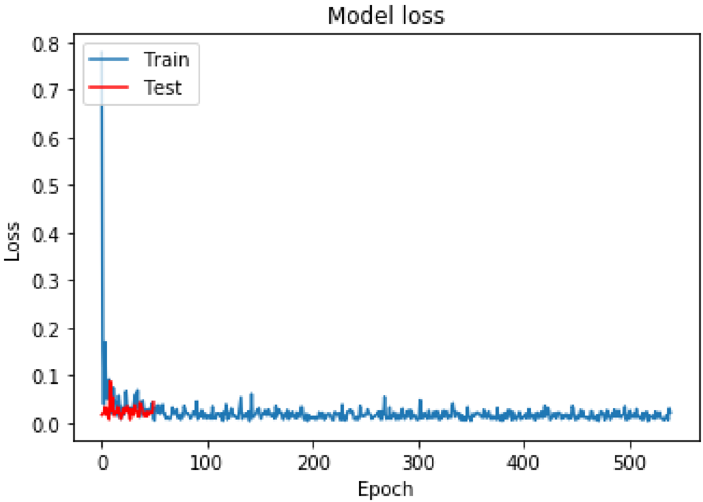



Fig11: Loss Function

# 3.5 Results

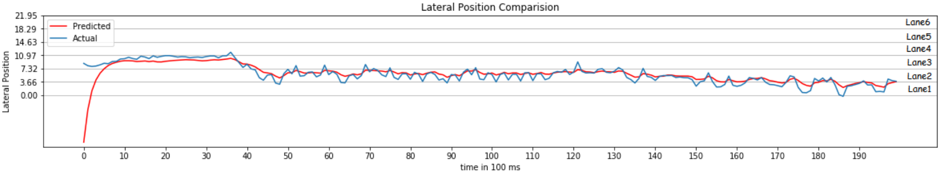



Fig12: Result1

# 3.5 Results

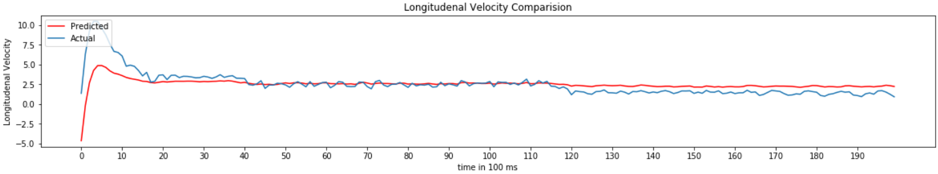



Fig13: Result2

# Outline

# 4. Conclusion

- our goal was to predict the future trajectory of the target vehicle. Since the region of interest spans roughly 1km longitudinally,
- The values of the longitudinal position can become quite large; for this reason, we prefer to predict future longitudinal velocities $\hat{v}_{y_{targ}}$ instead. Since the lateral position is bounded,
- we directly use $\hat{x}_{targ}$ for the output. In order to have different horizons of prediction, we choose a vector of outputs $\{\hat{x}_{k_{targ}};\ \hat{v}_{y_{k_{targ}}}\}$ for k=1:K consisting in values taken k seconds in the time.

To visualize the results in actual units the output of the neural network must be denormalized. To do this following calculations and values(from Table 1) are used:

Values used for Denormalization(for vehicle 595)

| S.No. | Quantity | Value |
|---|---|---|
| 1 | $x_{targ_{min}}$ | -28.43 m |
| 2 | $x_{targ_{max}}$ | 30.33 m |
| 3 | $\epsilon_{x_{targ}}$ | 10.10 m |
| 4 | $v_{y_{targ_{min}}}$ | -13.38 m/s |
| 5 | $v_{y_{targ_{max}}}$ | 38.07 m/s |
| 6 | $\epsilon_{v_{y_{targ}}}$ | 1.94 m/s |

# 4. Conclusion

The prediction is done for 20 seconds. Actual values are calculated from predictions using these formulas:

$$x_{targ_{actual}} = ((x_{targ_{normalized}} + \epsilon_{x_{targ_{normalized}}}) * (x_{targ_{max}} - x_{targ_{min}}) + x_{targ_{min}})$$

$$v_{y_{targ_{actual}}} = ((v_{y_{targ_{normalized}}} + \epsilon_{v_{y_{targ_{normalized}}}}) * (v_{y_{targ_{max}}} - v_{y_{targ_{min}}}) + v_{y_{targ_{min}}})$$

As visible in calculation by adding the average error value in the prediction the actual error remaining is very less, which is given in Table 2.

Final Errors(for vehicle 595)

| S.No. | Quantity | Value |
|---|---|---|
| 1 | $\epsilon_{x_{targ}}$ | -2.7755575615628914e-17 m |
| 2 | $\epsilon_{v_{y_{targ}}}$ | -5.662137425588298e-17 m/s |

# 4. Conclusion

- As we can see in Figure 12 and 13 the accuracy level is very high.Due to initialization with minimum values it takes almost 1 second for the predictions to reach minimum amount of accuracy after that the difference between signals are very less.
- In Figure 12 as we can see that the predicted lateral position is almost same as the actual values, Also to visualize the maneuvers y axis is divided based on lanes so that the lane changing maneuver can be visualized.
- In figure 13 we can see that the actual velocity of the vehicle and the predicted velocity are almost same. Also due to filtering the little changes in velocity is removed but still the the network predicts the velocity around the actual value with almost zero error.

# References I

[1] Florent Altché and Arnaud de La Fortelle. “An LSTM network for highway trajectory prediction”. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2017, pp. 353–359.

[2] ByeoungDo Kim et al. “Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network”. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2017, pp. 399–404.

[3] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. “A survey on motion prediction and risk assessment for intelligent vehicles”. In: *ROBOMECH journal* 1.1 (2014), p. 1.

[4] Christopher Tay, Kamel Mekhnacha, and Christian Laugier. “Probabilistic Vehicle Motion Modeling and Risk Estimation”. In: *Handbook of Intelligent Vehicles* (2012), pp. 1479–1516.

[5] Thomas Streubel and Karl Heinz Hoffmann. “Prediction of driver intended path at intersections”. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE. 2014, pp. 134–139.

[6] Ashwin Carvalho et al. “Stochastic predictive control of autonomous vehicles in uncertain environments”. In: *12th International Symposium on Advanced Vehicle Control*. 2014, pp. 712–719.

# References II


[7] Hiren M Mandalia and Mandalia Dario D Salvucci. "Using support vector machines for lane-change detection". In: *Proceedings of the human factors and ergonomics society annual meeting*. Vol. 49. 22. SAGE Publications Sage CA: Los Angeles, CA. 2005, pp. 1965–1969.

[8] Puneet Kumar et al. "Learning-based approach for online lane change intention prediction". In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2013, pp. 797–802.

[9] Adam Houenou et al. "Vehicle trajectory prediction based on motion model and maneuver recognition". In: *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2013, pp. 4363–4369.

[10] Seungje Yoon and Dongsuk Kum. "The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 1307–1312.

[11] Aida Khosroshahi, Eshed Ohn-Bar, and Mohan Manubhai Trivedi. "Surround vehicles trajectory analysis with recurrent neural networks". In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2016, pp. 2267–2272.

# References III


[12] Derek J Phillips, Tim A Wheeler, and Mykel J Kochenderfer. "Generalizable intention prediction of human drivers at intersections". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 1665–1670.

[13] Assaad MOAWAD. *Neural networks and back-propagation explained in a simple way*. https://medium.com/datathings/neural-networks-and-backpropagation-explained-in-a-simple-way-f540a3611f5e. 2018.

[14] Matt Mazur. *A Step by Step Backpropagation Example*. https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example. 2015.

[15] Michael Nguyen. *Illustrated Guide to LSTM's and GRU's: A step by step explanation*. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21. 2018.

[16] Shi Yan. *Understanding LSTM and its diagrams*. https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714. 2016.

[17] Colah's Blog. *Understanding LSTM Networks*. https://colah.github.io/posts/2015-08-Understanding-LSTMs. 2015.

# References IV


[18] U.S. Federal Highway Administration. *US Highway 101 dataset.* https://catalog.data.gov/dataset/next-generation-simulation-ngsim-vehicle-trajectories. 2005.