Chair for Applied and Numerical Analysis

Maneuver prediction using vehicle sensor data

TUM Data Innovation Lab

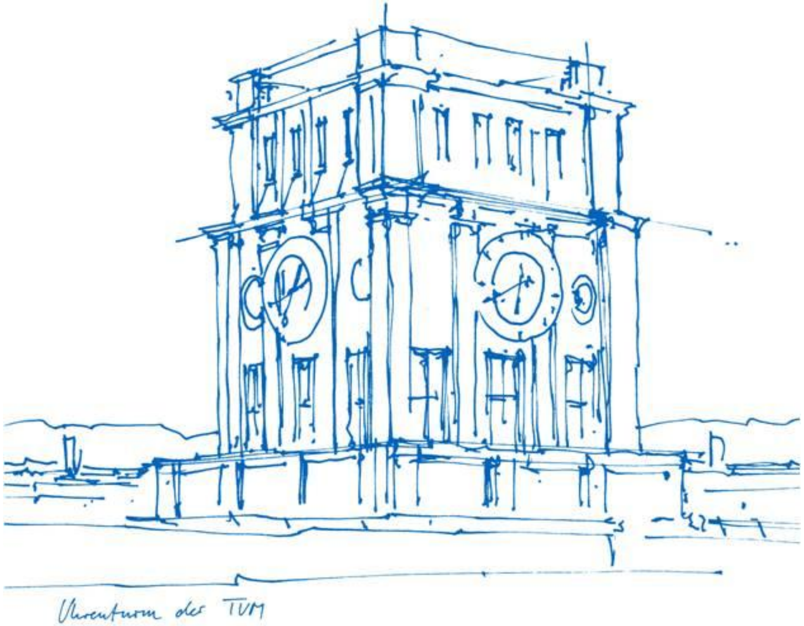
Team:
Nils Sturma, Harshit Chopra, Anna Kuvakina

Mentors:
Nico Epple,
Dr. Benny Kneissl

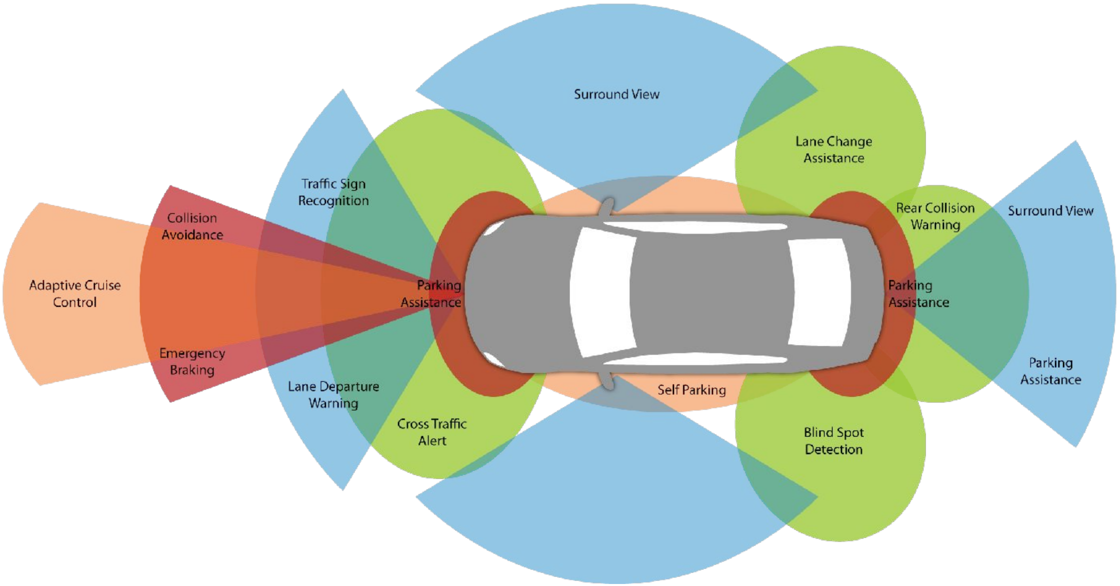
Co-Mentor:
Laure Vuaille

Project Lead:
Dr. Ricardo Acevedo Cabra

Supervisor:
Prof. Dr. Massimo Fornasier



Motivation



Range of functionality of ADAS systems [1]

“We focus on the intention of the driver and develop an approach for lane change prediction”

Naturalistic driving studies

Test vehicles collect real world data in different countries using

- multiple sensors
- front and rear view cameras

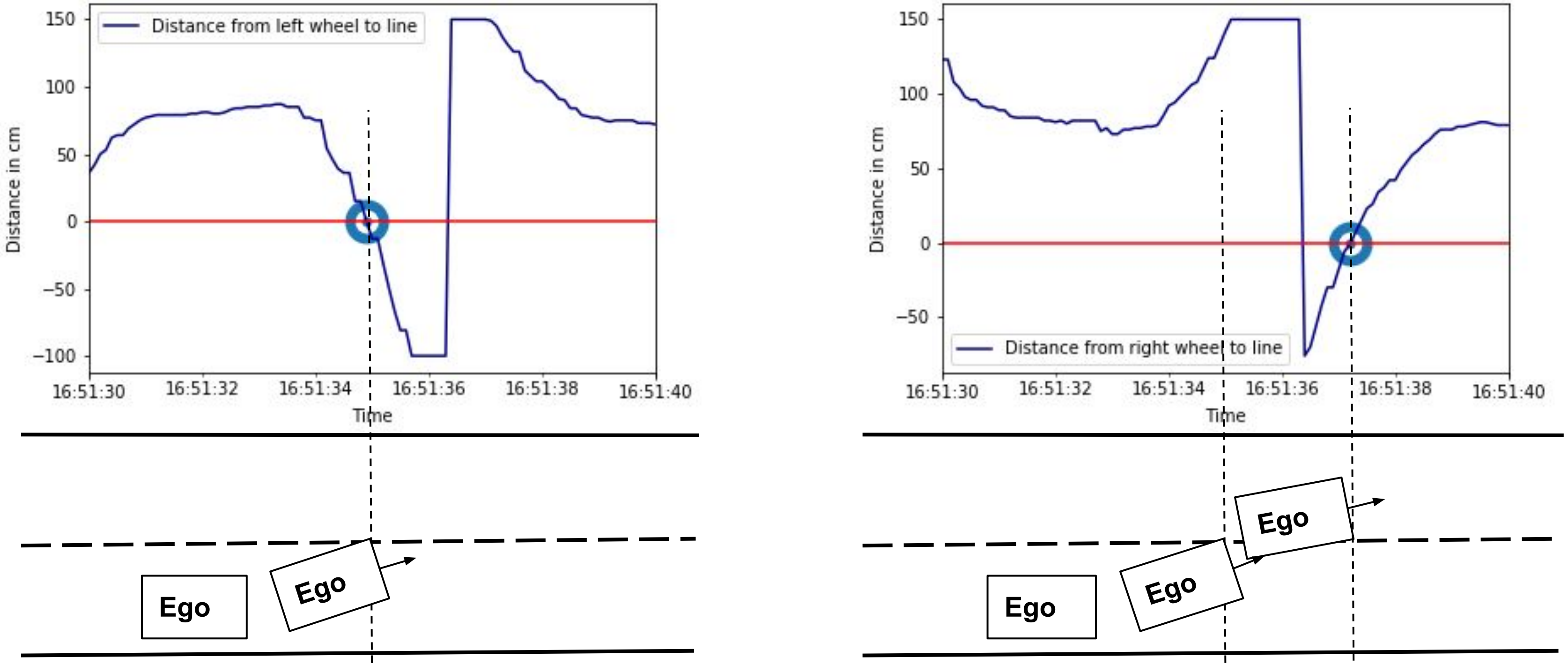
We work with **signal** and **object** data collected on German highways

Overview

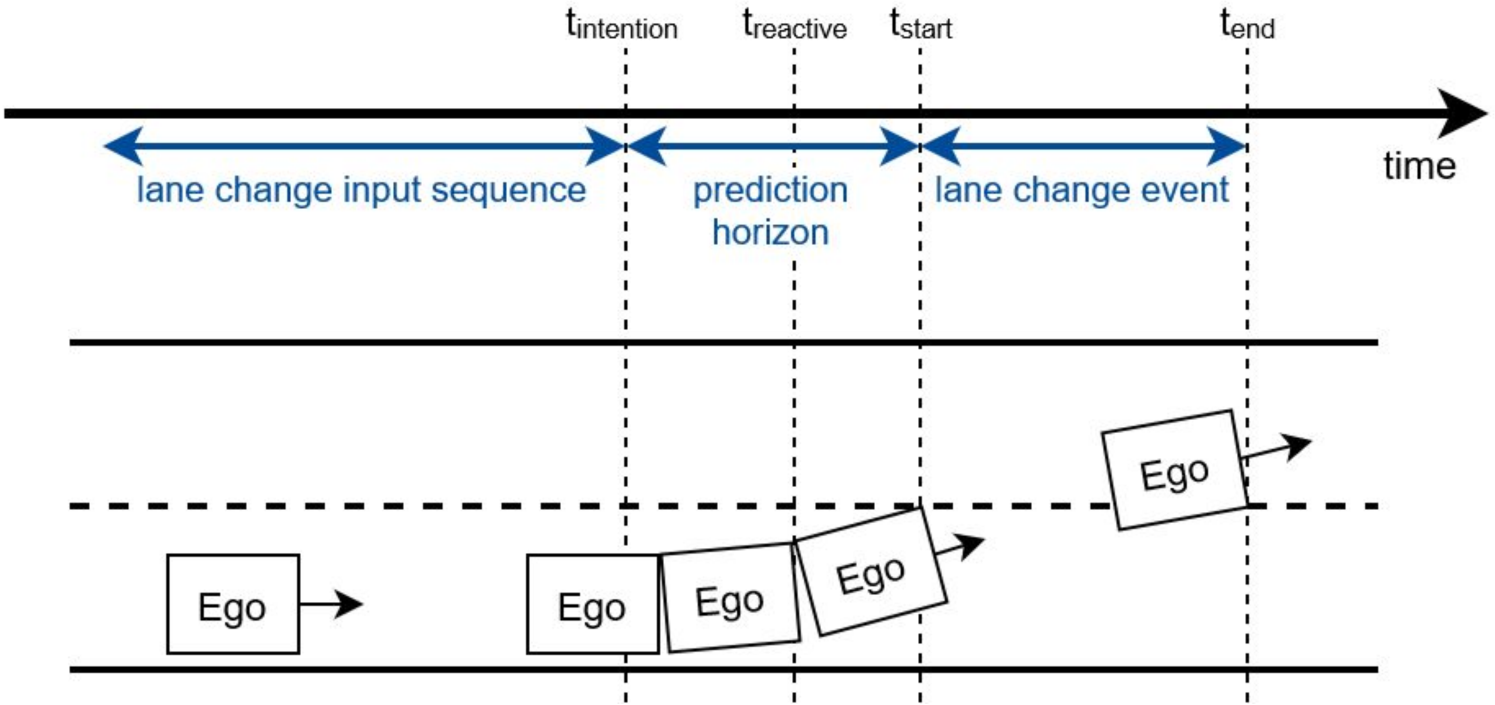
- Preprocessing
- Feature engineering
- Data exploration
- Algorithms
- Results
- Case studies

Labeling

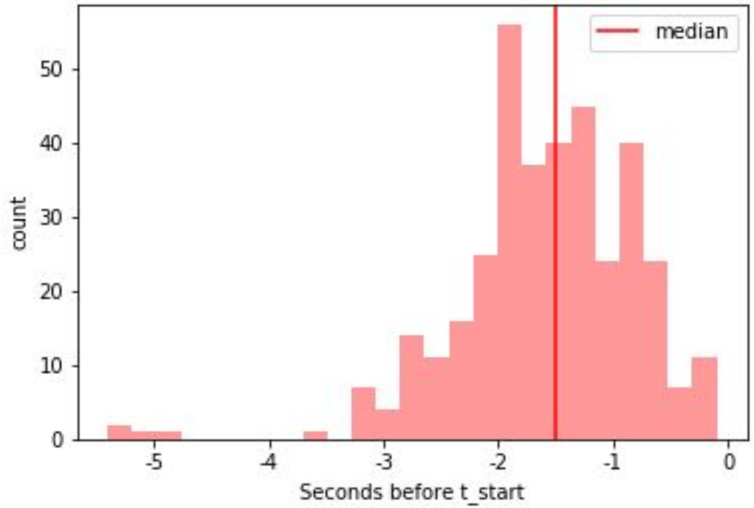
- automatic, rule-based algorithm
- validation: videos
- ground truth



Prediction horizon



Prediction horizon

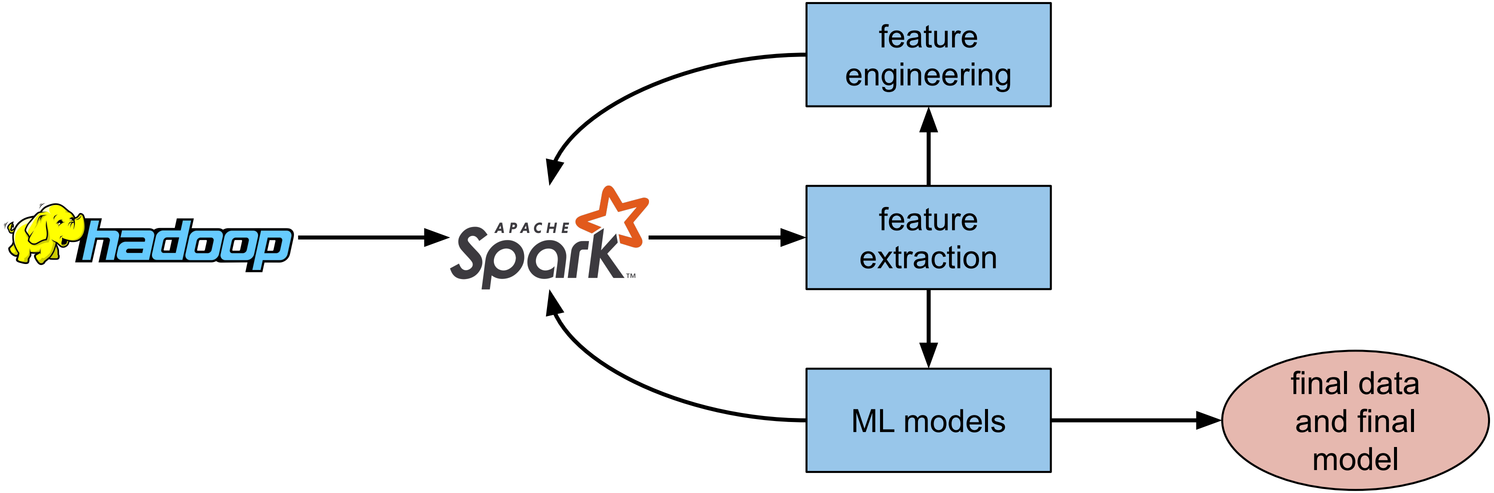


Used features
- situation
- kinematics
- surrounding objects

NOT used features
- winker
- distance to line left/right
- steering wheel angle
- lateral acceleration

Goal: predict driver's intention

Data workflow



	Train	Test
lane change samples	13416	887
non lane change samples	68393	4113

Overview

Preprocessing

Feature engineering

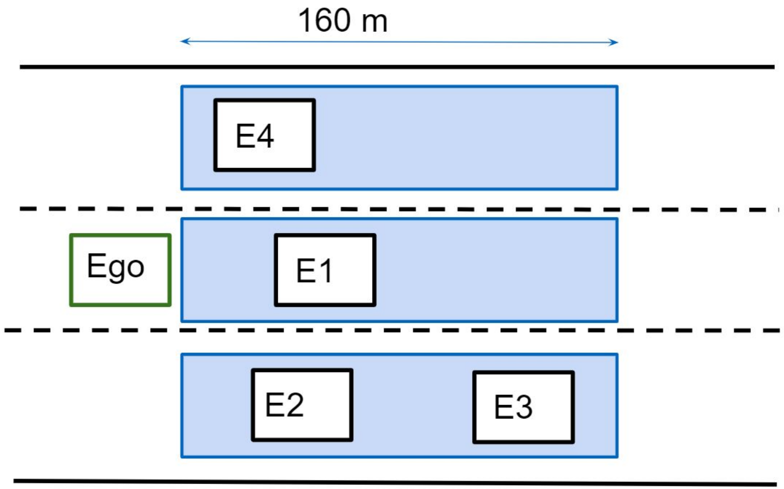
Data exploration

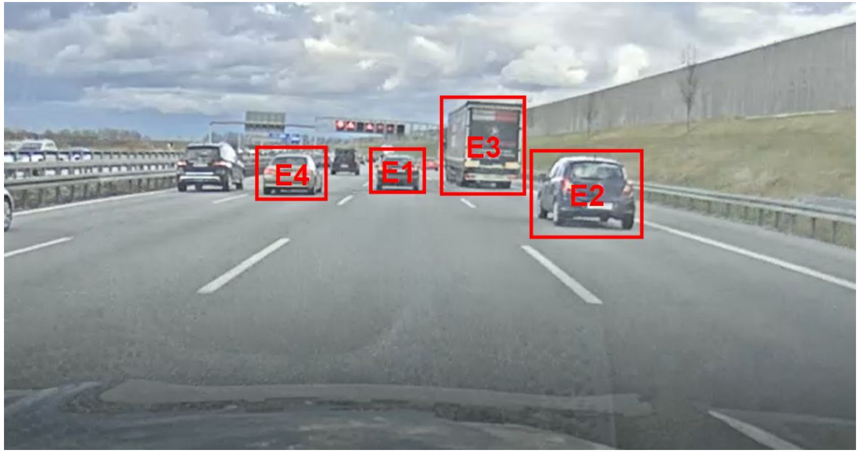
Algorithms

Results

Case studies

Surrounding objects





Features

Velocity

Lane count

Acceleration

Blind spot detection (left and right)

Road angle

Boundary line type (left and right)

Features

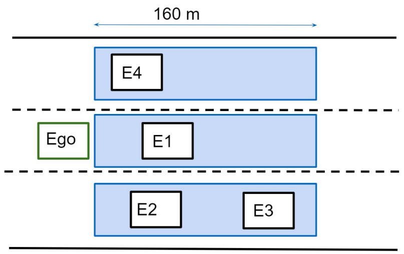
- Velocity
- Lane count
- **Absolute velocity** of surrounding objects
- Acceleration
- Blind spot detection (left and right)
- **Distance** to surrounding objects

- Road angle
- **Absolute acceleration** of surrounding objects
- Boundary line type (left and right)

Feature engineering

- Inverse time to collison

$$\frac{speed\_Ego - speed\_E1}{distance\_Ego\_E1}$$



- Time headway

$$\frac{distance\_Ego\_E1}{speed\_Ego}$$

- Relative speed

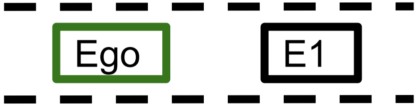
- Handling missing objects

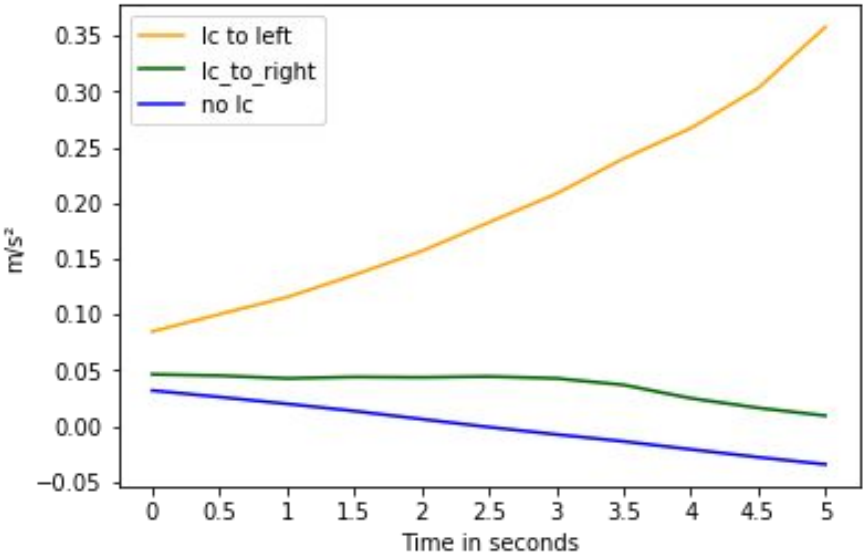
E1 exists	E2 exists	E3 exists	E4 exists
yes/no	yes/no	yes/no	yes/no

Overview

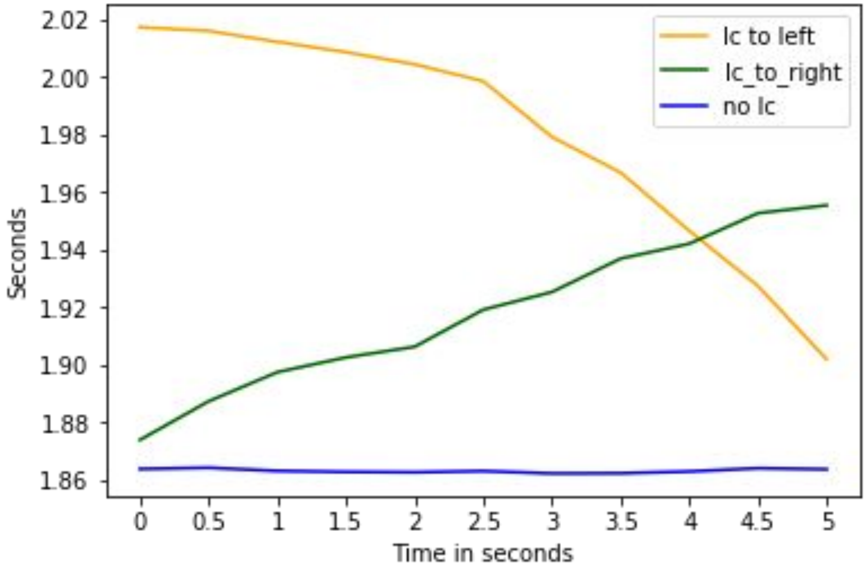
- Preprocessing
- Feature engineering
- **Data exploration**
- Algorithms
- Results
- Case studies

Continuous features



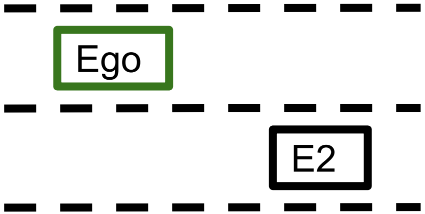


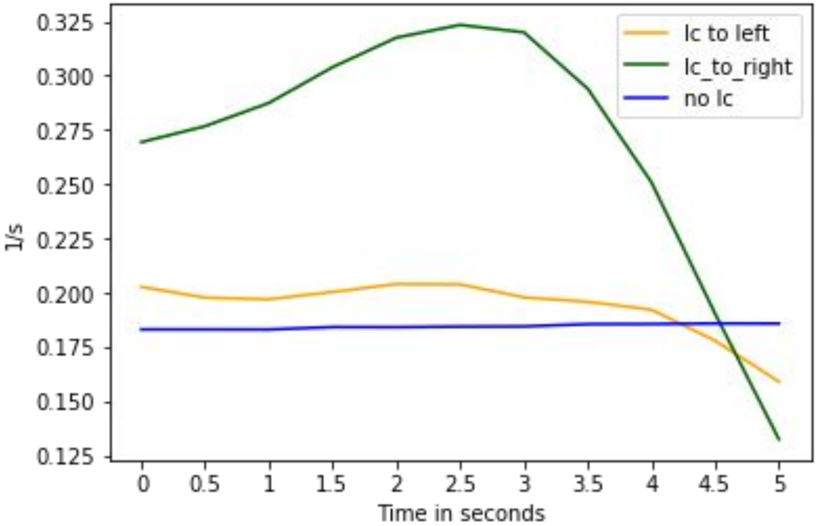
Mean acceleration of Ego



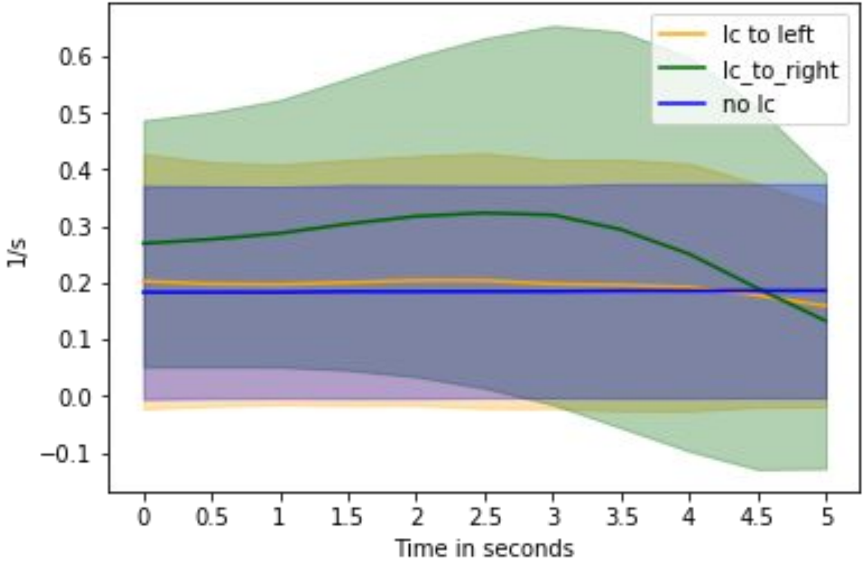
Mean time headway between Ego and E1

Continuous features



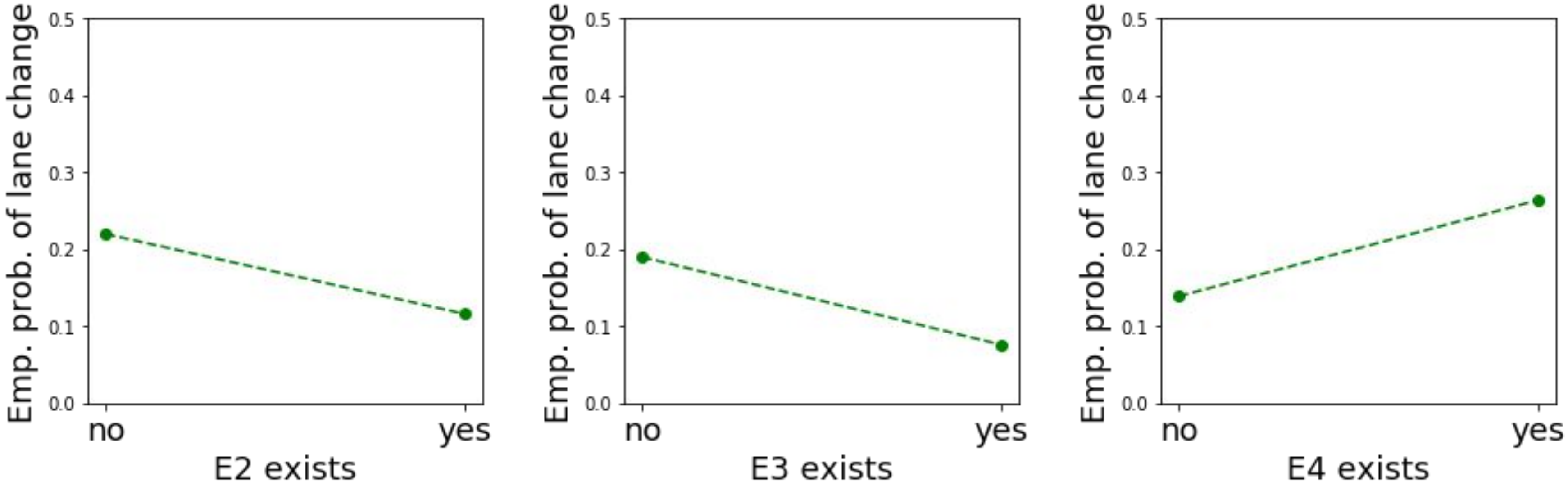


Mean inverse time to collision between Ego and E2



Mean inverse time to collision between Ego and E2 with std. deviations

Categorical features



Probability of lane change conditioned on the existence of vehicles

Overview

- Preprocessing
- Feature engineering
- Data exploration
- **Algorithms**
- Results
- Case studies

Performance metrics

TP

TN

FP

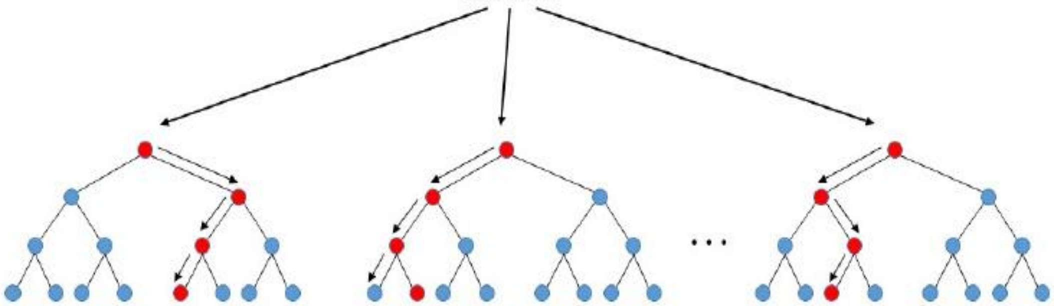
FN

$$precision = \frac{TP\_count}{TP\_count + FP\_count}$$

$$recall = \frac{TP\_count}{TP\_count + FN\_count}$$

$$F_1 = \frac{2 * precision * recall}{precision + recall}$$

Random Forest



Random Forest [6]

$$G = 1 - p_{lc}^2 - p_{nlc}^2$$

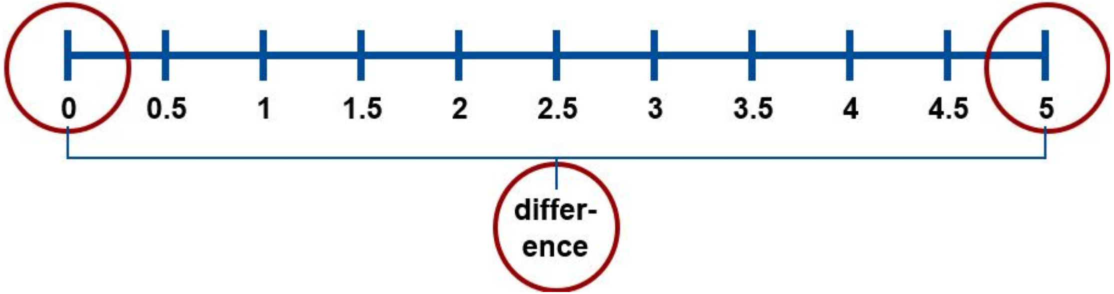
$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

Random Forest

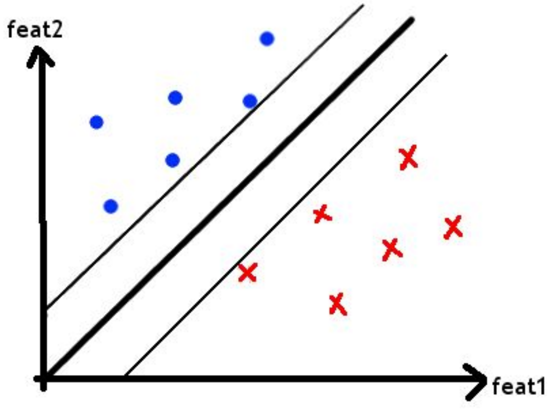
- no scaling
- fast training

F1 score	Accuracy
0.6864	0.7946

Used features:



Support Vector Machine



Linear SVM classifier [2]

- Same features as for Random Forest
- Kernel Trick (RBF kernel)
- Grid Search
  - Gamma
  - Penalty parameter
- Forward Feature Selection
- Backward Elimination

Support Vector Machine

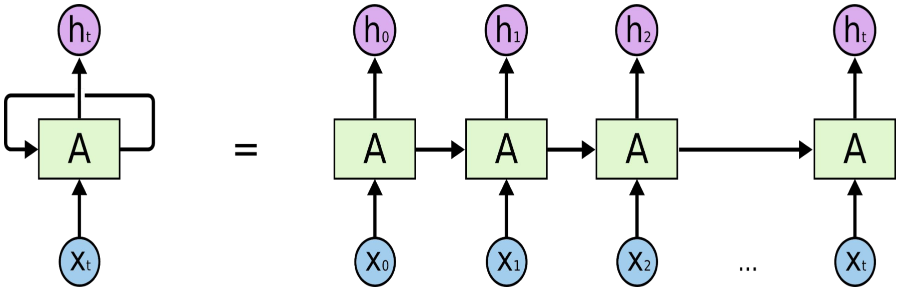
Forward Feature Selection Algorithm

Best Features	F1 score
vehicles right count	0.3659
boundary line left (type 3)	0.3823
longitudinal velocity	0.4018
longitudinal acceleration	0.4406
difference of velocity between Ego and E2	0.4863
boundary line left (type 3) start	0.4977
inverse time to collision between Ego and E1	0.5107

F1 score	Accuracy
0.5642	0.8273

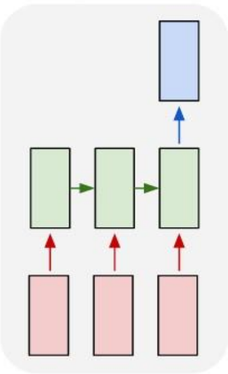
Recurrent Neural Networks

- Learn relations between sequences of inputs
- Input samples - (no_of_timestamps * features)
- LSTM/GRU
- Loss: Binary cross entropy



Unrolling in RNN [3]

$$-\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$$

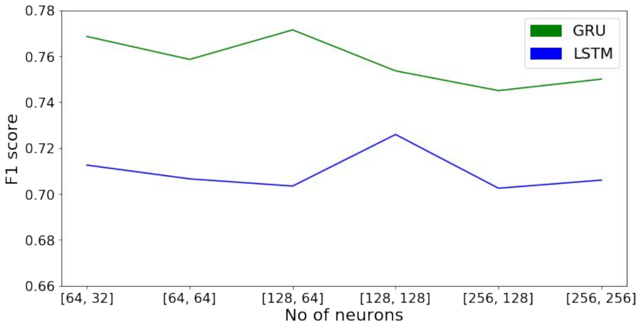


Many-to-one RNN network [4]

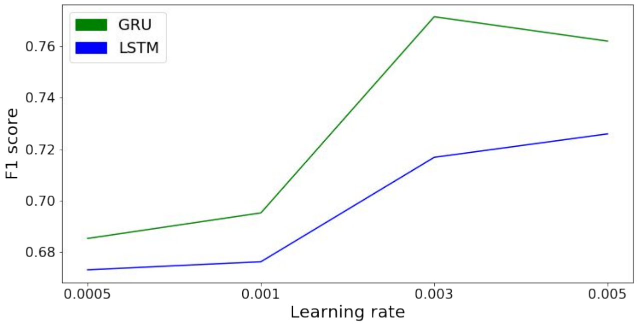
Recurrent Neural Networks

- Hyperparameter tuning
- Optimizer - RMSProp
- Other network features
  - Dropout
  - Input weight sampling
  - Weight regularization
  - Reduce learning rate
- Final GRU structure:

Recurrent 1	128
Recurrent 2	64
Dense 1	32
Dense 2	1



F1 score variation with different number of neurons in recurrent layers

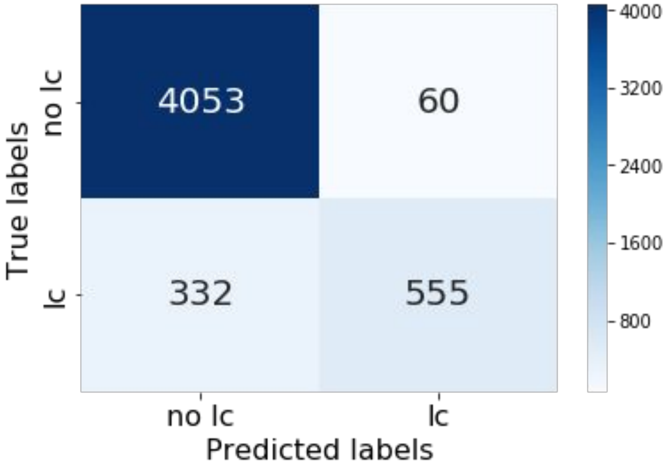


F1 score variation with initial learning rate

Overview

- Preprocessing
- Feature engineering
- Data exploration
- Algorithms
- **Results**
- Case studies

Results



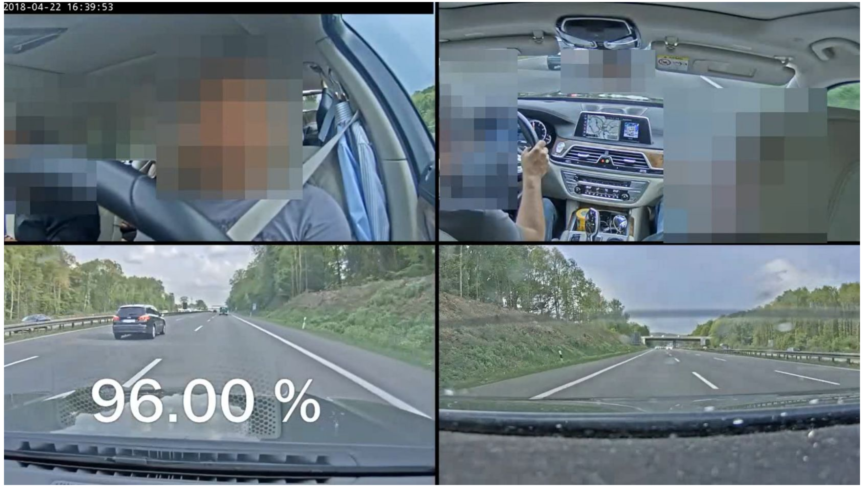
F1 score	Accuracy
0.7714	0.9264

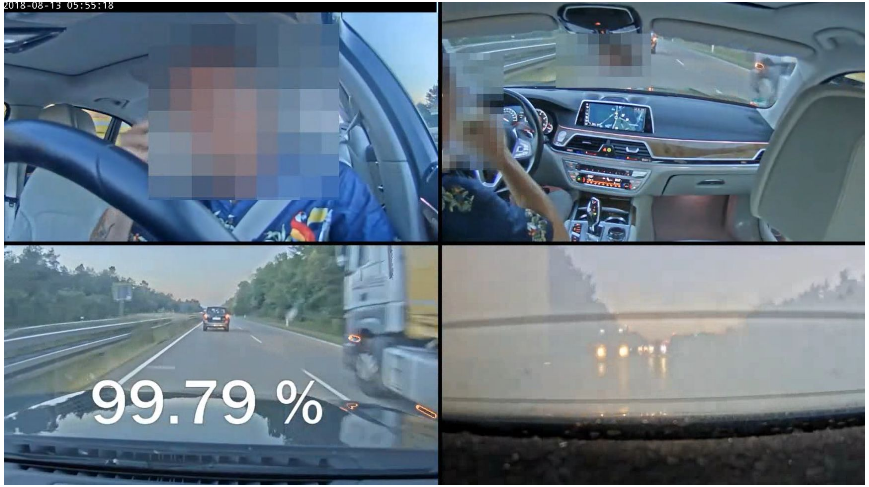
Lane changes	887
Non lane changes	4113

Model which predicts direction of lane change:

F1 score	Accuracy
0.7470	0.9014

Visualization





Edge cases

TN	FP
FN	TP

Driver overtakes and does not go right again

TN	FP
FN	TP

Driver changes lane to left with no objects around

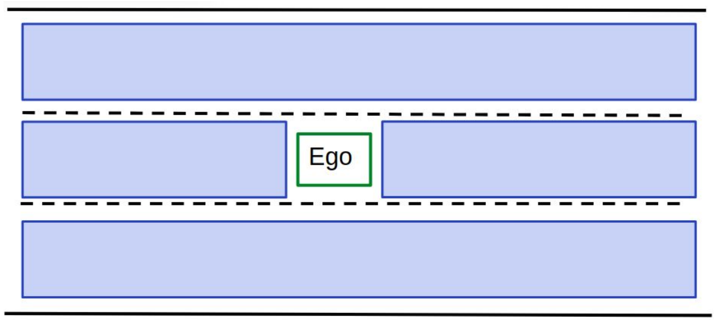
Further improvements

Clustering of drivers according to driving behavioural patterns

Clustering of people [6]

Data improvements

- Self-representation of Ego
- Vehicles behind Ego



Overview

- Preprocessing
- Feature engineering
- Data exploration
- Algorithms
- Results
- **Case studies**

Case studies

China data

- Transfer our trained model directly
- Test transferability to geographically different environment

All features

- Train new model using features that were excluded like winker, distance to line, lateral acceleration
- Assert improvement by these features

Experiment	F1 score	Accuracy
Direct transfer China	0.2664	0.6276
Using all features	0.8828	0.9366

Conclusion

automatic rule-based labeling ✓

feature engineering on large scale ✓

lane change prediction ✓

Transferable?

- model (direct) ☐
- approach ✓

explore possible further improvements ✓

References

[1] https://www.kisspng.com/png-car-advanced-driver-assistance-systems-driving-veh-3194115/preview.html [visited on 20/07/2019]

[2] https://michelleful.github.io/code-blog/2015/06/18/classifying-roads/ [visited on 30/07/2019]

[3] https://colah.github.io/posts/2015-08-Understanding-LSTMs/ [visited on 03/08/2019]

[4] https://discuss.pytorch.org/t/example-of-many-to-one-lstm/1728 [visited on 28/07/2019]

[5] https://towardsdatascience.com/k-means-clustering-identifying-f-r-i-e-n-d-s-in-the-world-of-strangers-695537505d [visited on 03/08/2019]

[6] https://www.linkedin.com/pulse/random-forest-algorithm-interactive-discussion-niraj-kumar/ [visited on 30/07/2019]

Thank you for your attention!

Questions?