Prediction Module

Assignment

Prediction Approaches
- Model-Based
- Data-Driven

Lane-Sequence-Based Prediction
- Predict target lane
- Draw lane-based trajectory

Neural Networks

Recurrent Neural Networks

RNNs for Target Lane Prediction
- Obstacle status feature and lane feature
- Network Architecture

Prediction Module

Lane-Sequence-Based Prediction

Prediction is a common component of IQ tests such tests often have question in the form of what comes next 1 1 2 3 5 8 13 and then the test has to fill in the blanks.

1 1 2 3 5 8 13 [ ] [ ]

Prediction problem is frequently implemented by agents typically possess the feature of incrementally predicting the evolution of a continuous stream of sensory data. agent in a way could exploit learned data to explore the future.

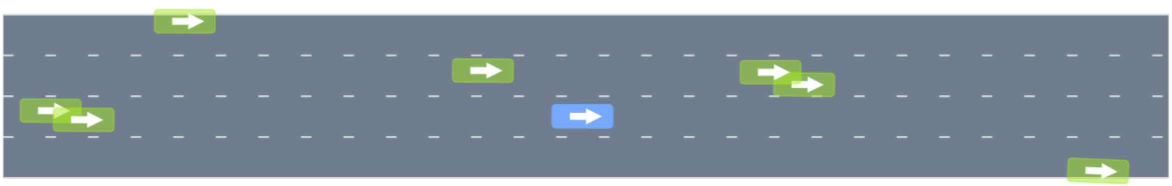
Predicting other traffic participants trajectories is a key task for an autonomous vehicle, in order to avoid collisions on its planned trajectory.

The prediction module receives the obstacles information e.g. including their positions, headings, velocities, accelerations, and generates the predicted trajectories with probabilities for the obstacles. In short, first predict the behavior and then computes the actual trajectory.

Moving behavior of road objects is dependent on the the type of object. A vehicle's behavior might be *keep lane* or *change lane* on highway*, make a turn* in urban environment. While cyclists or pedestrians have totally different possible behaviors.

We can classify the behavior by enumerating all possible behaviors but not really. For example, there could be just one lane, or multiple left and right-turn lanes, or even free moving obstacles block all possible lanes. So, we can not choose a proper categories of behaviors based on distinctive maps as a lot of "labels" will lead to an overwhelmingly complicated and not scalable situation.

Lane-Sequence is one of the above mentioned scenarios in which vehicle objects move along lanes in a multiply lane landscape thus we impose an assumption that vehicle follows certain logical or structural sequence of lanes upon which behavior is easy to predict.



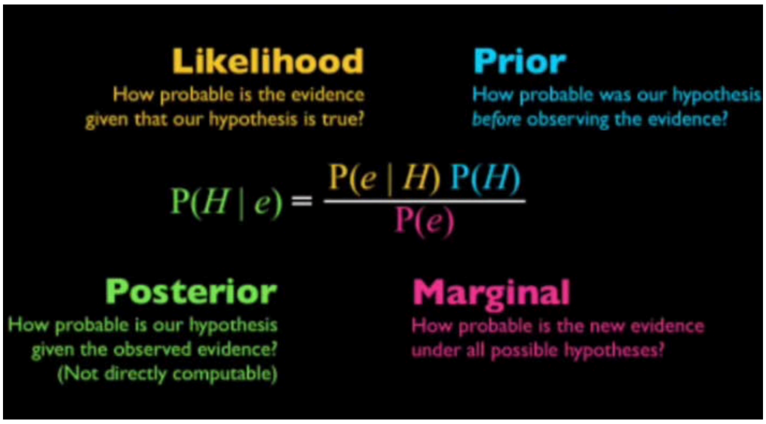
In order to predict the behavior of exo-vehicle and find possible trajectory of the ego-vehicle, on each possible lane a binary classification problem solver could apply. i.e. it is a target lan e or not for each candidate lane sequence.

Prediction Module

Predict Target Lane

While detecting other vehicles and localizing them to the map, we focus on developing a framework that uses proper object features to predict near future target lane probabilities.

say, we calculate path and speed separately(observations) for any given obstacle and output a probability of maneuver towards a target lane. such as performing left or right lane change or staying in the same lane.



According to bayes rule, The equation to compute the probability of any maneuver Mt can be written as:

$$P(M_t|Z_{0:t}) \propto P(Z_t|M_t) \times \left[\sum_{M_{t-1}=L,R,N} P(M_t|M_{t-1}) \times P(M_{t-1}|Z_{0:t-1}) \right]$$

Zt corresponds to the observations at time t ,The posterior P(Mt|Z0:t) is used for the final lane change intention prediction.

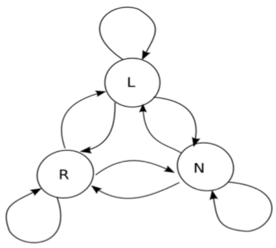
The likelihood term P(Zt |Mt), is generated by a way of Bradley-Terry Model. Bradley–Terry model is a probability model that can predict the outcome of a comparison. Given a pair of individuals i and j drawn from some population, it estimates the probability that the pairwise comparison i>j turns out true, as

$$P(i > j) = \frac{p_i}{p_i + p_j}$$

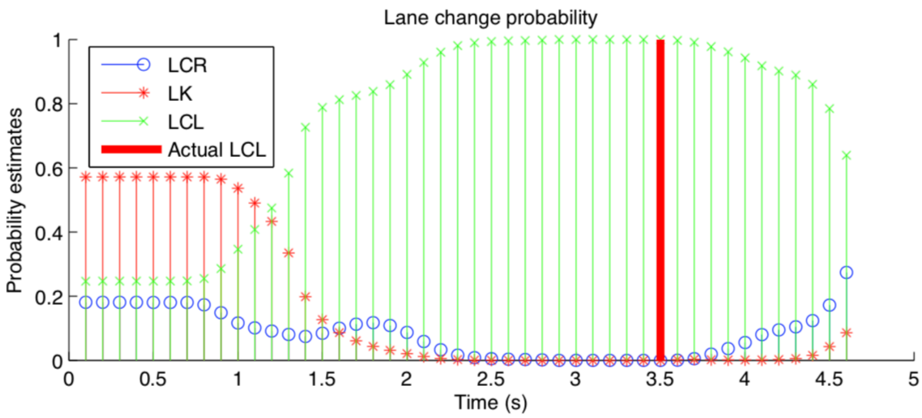
By using of a probabilistic outputs from the multiclass SVM algorithm. This approach extends the Bradley-Terry model for paired individual comparisons so that multi-class probability estimates can applied. Thus we can compare the likelihood of each possible behavior quantificationally.

Prediction Module

In prior term, L, R, N represent left, right and no lane changes respectively.The state transition probabilities P(Mt |Mt −1) are learned offline using training data. For all the three maneuvers(L,R,N) we get a 3 × 3 state transition matrix with 9 state transition probabilities.



We build a probability ranking method over all possible lane sequences to maximize the predictive power for exo-vehicle’s behavior towards their corresponding target lane.

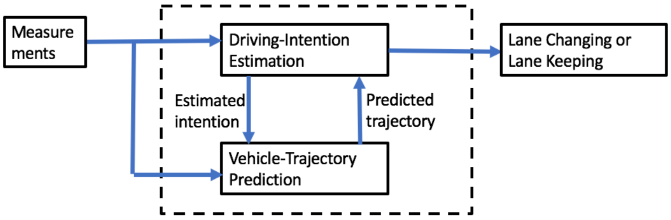


Prediction Module

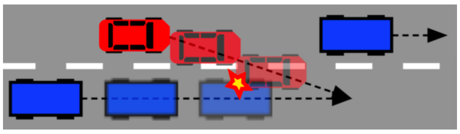
Draw Lane-based trajectory

Now we shall compute actual lane-based trajectory for a target vehicle towards its predicted driving goal while avoiding collision with adjacent vehicles. If a collision occurs during a lane change, the predicted trajectory is re-planned.

Driving-intention estimation module works with vehicle-trajectory prediction module together to decide if a lane changing or lane keeping approach would apply.

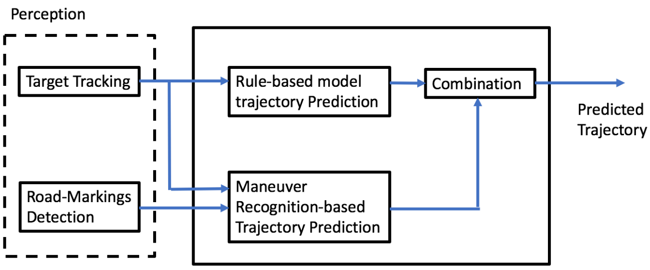


We assumes the adjacent vehicle(blue car) drives with a constant velocity during the prediction time. even if adjacent vehicles are accelerating or decelerating, this action is prominently reflected soon in the next time step as the prediction step performed at each time step. this update of position is conducted to check the possibility of collision between target (red car) and adjacent vehicles.



If collision forecasted during a lane change, the predicted trajectory is initialized and re-planned. Driving intention will be updated as *keeping*, even if the estimated driving intention was *changing*. This strategy can be explained as the abortion of a lane change by a driver when he feels unsafe because of the insufficient gap or velocity. It is also expected to eliminate false alarms caused by zigzag driving.

We could create trajectory prediction method in rule-based motion model or machine learning-based maneuver recognition approach. Or combine them to take benefit on the short-term accuracy of the first approach and the better accuracy of the second approach at longer term.



There are cases where the 2nd approach uses no training data, maneuver recognition can based on the comparison between target vehicle position and detected road-markings.

Prediction Module

Recurrent Neural Networks

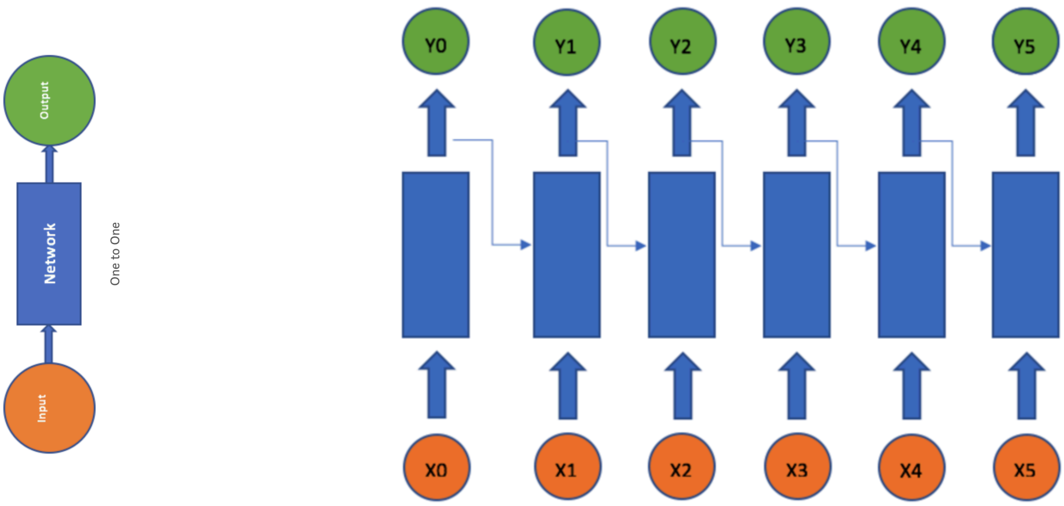
The simplest machine learning problem involving a sequence is a one to one problem. we have one data input to the model and the model generates a prediction with the given input.

One to One

A recurrent neural network(RNN) could deals with sequence problems too.In below graph, we aligned several one-to-one network in parallel. But one more route attached to each adjacent networks: it retains the output state from the former network and use it as input for current network and so on to the next.

In programming terms this is like running a fixed program with certain inputs plus the retained output as its internal variables. This structure indicates that the network can remember previous input data, and uses this in combination with latest input data to make new predictions.

RNN retain information from last time series iteration, in a way it "remembers". This memory is called the *state*.



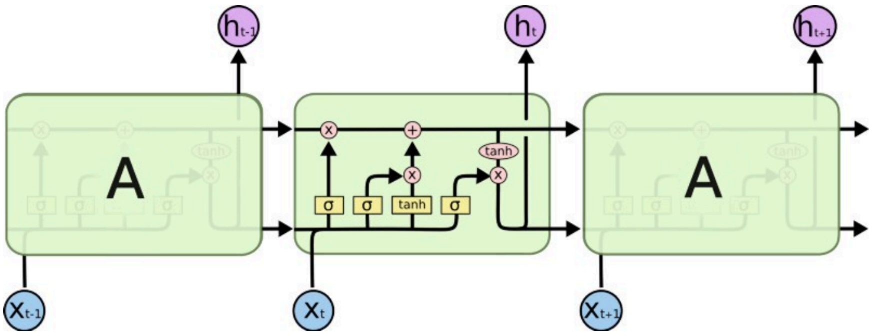
In this graph case only two weights are involved. The weight multiplying the current input *xt*, which is *u,* and the weight multiplying the previous output *yt-1*, which is *w*. Due to the recurrent nature, same parameters are used for all input points.

$$Y_t = \tanh(wY_{t-1} + u\,x_t)$$

Prediction Module

A simple recurrent network suffers from a fundamental problem of not being able to capture long-term dependencies in a sequence. one popular variant of RNN is long short term memory (LSTM) model, which effectively overcomes the issue in naively designed RNNs.

The LSTM consists of the cell memory that stores the summary of the past input sequence, and the gating mechanism by which the information flow between the input, output, and cell memory are nicely controlled.

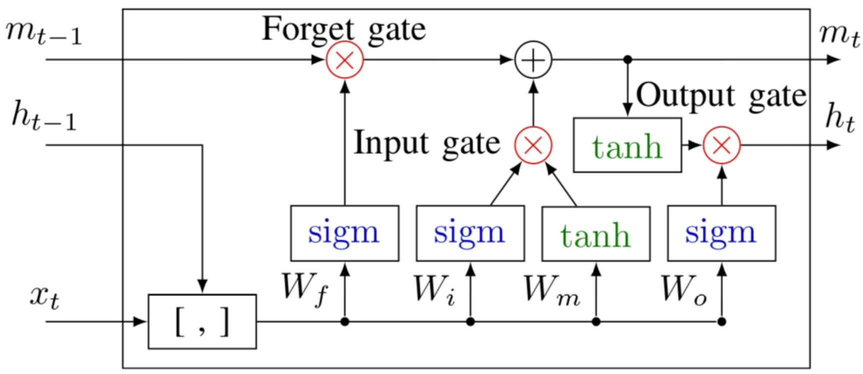


LSTM Architecture

An interesting feature of LSTM cells is the presence of an internal state which serves as the cell's memory, denoted by mt. Based on a new input xt, its previous state mt−1 and previous output ht−1, the cell performs different operations using so-called ***gates***:

• ***forget gate*:** uses the inputs to decide how much to "forget" from the cell's previous internal state(memory) mt−1;
• ***input gate*:** decides the amount of new information to be stored in memory based on xt and ht−1;
• ***output gate*:** computes the new cell output from a mix of the previous states and output of the input gate.

This particular feature of LSTM allows a network to learn long-term relations between features, which makes them very powerful for time series prediction.



Prediction Module

RNNs for Target Lane Prediction - Obstacle status feature and lane feature

Feature engineering and design is a critical aspect of building a vehicle behavior predictor. In our case, we could consider 2 categories of possible features: obstacle feature and lane feature.

We aim at only using features which can be reasonably easily measured using on-board sensors like GNSS and LiDAR/Radar. we consider a set of features for the target vehicle for which we want to compute the future trajectory.

target tracking system hosted by the ego-vehicle provides for each target vehicle the state vector $\zeta^{(n\text{th target})}$, Below features are considered for obstacles: longitudinal and lateral coordinates x and y; longitudinal and lateral velocity $\dot{x}(n)$, $\dot{y}(n)$.

$$\zeta^{(n\text{th target})} = [x(n),\ y(n),\ \dot{x}(n),\ \dot{y}(n)]$$

For prediction operation, the working frame is static and corresponds to the local measurement frame. the ego-vehicle's state vector is thus defined as $\zeta^{(\text{ego})}$: where v, $\dot{\psi}$ velocity and yaw rate are provided by proprioceptive sensors.

$$\zeta^{(\text{ego})} = [v,\ \dot{\psi}]$$

Along the longitudinal direction of the target lane central reference, we can extract below lane features: longitudinal and lateral position of center lane points x and y; lane heading θ; the curvature of lane γ, The estimated value of γ is obtained from below equation, where v and ω ($\dot{\psi}$) are the velocity and the yaw rate of the vehicle.

$$\gamma = \frac{\omega}{v}$$

Prediction Module

RNNs for Target Lane Prediction - Network Architecture

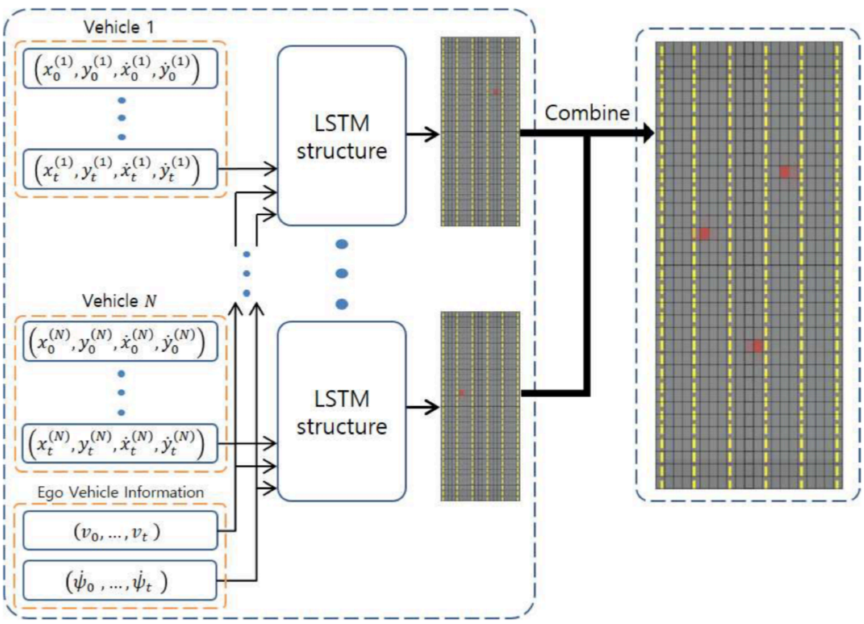
Contrary to many existing frameworks to predict categories of vehicle behavior, which can be solve in a classical probabilistic modeling approach. our aim is to predict future (x, y) positions for the target vehicle, we choose to use the recurrent neural network(RNN) for our learning architecture, which is particularly well suited for time series.

Due to the recurrent nature, even a single layer of RNN nodes can iterate over time to form a "deep" neural network. the role of the RNN layer is to abstract a meaningful representation of the input time series then output the predicted future states like vehicle trajectory.

The set of the observations used for prediction of the trajectory of the *n*th surrounding vehicle at the time step t is given by:

$$O_t^{(n)} = \{v_t, \dot{\psi}_t, x_t^{(n)}, y_t^{(n)}, \dot{x}_t^{(n)}, \dot{y}_t^{(n)}\}$$

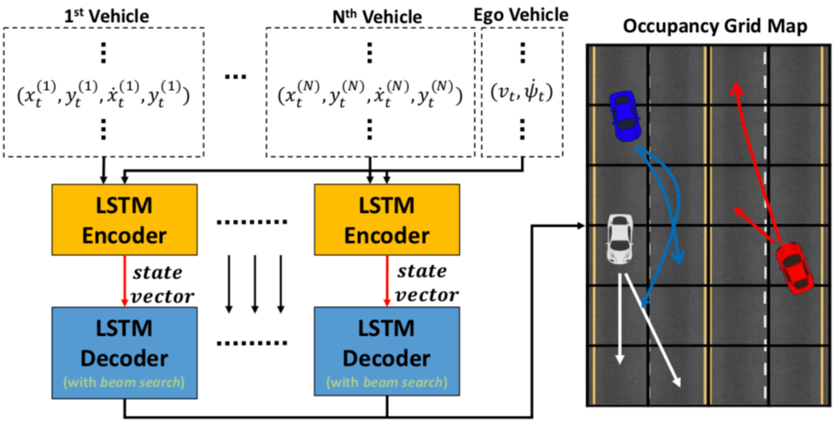
Where we included the ego-vehicle's speed and yaw rate and the *n*th exo-vehicle's relative coordinate/velocity. Which will feed to the RNN(actually LSTM) network time series.



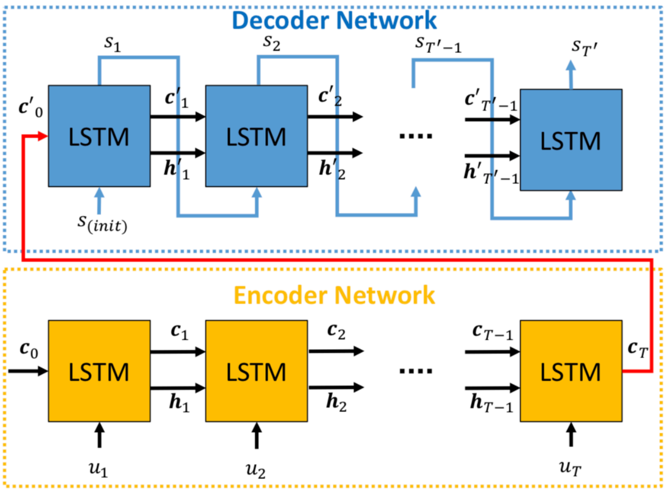
The proposed trajectory prediction system inputs the coordinates and velocities of the surrounding vehicles obtained from the sensor measurements to the LSTM and produces the vehicle's future location after Δ seconds. The LSTM is designed to produce the probability of occupancy for the surrounding vehicles on the OGM map which we will introduce later.

Prediction Module

Recent studies introduced the LSTM encoder and decoder architecture for the trajectory prediction task.



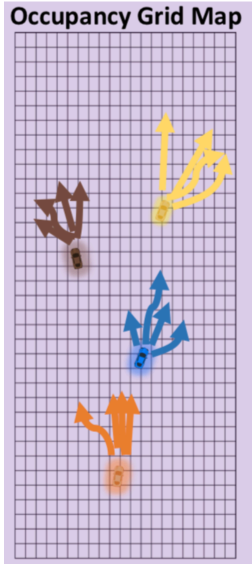
This structure has a name: *Sequence to Sequence*. we connected 2 RNNs in series. First RNN compress the sequence input (length M) to a context vector C_T (length 1), Second RNN generate output (length N) from the context vector C_T. In short, it's a M >> N prediction model which can handle input/output sequence at any indefinite length.



The LSTM encoder takes the latest trajectory samples for the surrounding vehicles as well as the state information on the ego-vehicle and produces the fixed-length vector(C_T) which captures the key context of the past trajectory. Based on the fixed-length vector, the LSTM decoder generates the future trajectory onto the map.

Prediction Module

In order to represent the future trajectory of the surrounding vehicles predicted by the proposed system, we use the occupancy grid map (OGM) which has been widely used for the object localization.



The OGM divides the region around the ego vehicle into Qw × Ql rectangular grid elements. a single grid will cover the length of a typical sedan and about quarter of the lane width.

Using the OGM, the trajectory prediction task becomes the classification problem of choosing one of Q = Qw ×Ql +1 classes corresponding to Qw ×Ql grid elements of the OGM and the out-of-map state. Therefore, we formulate the vehicle trajectory prediction as a sequential multi-class classification problem where the grid element occupied by a surrounding vehicle should be chosen sequentially for each time step.

Finally, we can align the lane features with the OGM. To map the predicted trajectory onto global coordinates in the real world and vice versa.

Prediction Module

References

Comparative Evaluation of Occupancy Grid Mapping Methods Using Sonar Sensors	2:26 PM
Occupancy Grid Maps for Localization and Mapping	10:50 AM
Sequence to Sequence Learning with Neural Networks	10:37 AM
Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation	9:15 AM
DESIRE- Distant Future Prediction in Dynamic Scenes with Interacting Agents	9:12 AM
Probabilistic Vehicle Trajectory Prediction over Occupancy Grid Map via Recurrent Neural Network	9:10 AM
Deep Tracking- Seeing Beyond Seeing Using Recurrent Neural Networks	9:08 AM
Long short term memory 1997	9:04 AM
Analysis of Recurrent Neural Networks for Probabilistic Modeling of Driver Behavior	8:45 AM
Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture	8:45 AM
MODELS FOR PEDESTRIAN TRAJECTORY PREDICTION AND NAVIGATION IN DYNAMIC ENVIRONMENTS	8:32 AM
PggISCI14	12:51 AM
Predicting trajectories of golf balls using recurrent neural networks	12:45 AM
trajprediction	Yesterday
Vehicle Trajectory Prediction based on Motion Model and Maneuver Recognition	Yesterday
Models Supporting Trajectory Planning in Autonomous Vehicles	Yesterday
Vehicle_trajectories_prediction_based__on_driver_behaviour_model	Yesterday
Graphical Models for Driver Behavior Recognition in a SmartCar	Yesterday
Generalized Bradley-Terry Models and Multi-Class Probability Estimates	Yesterday
Using Support Vector Machines and Bayesian Filtering for Classifying Agent Intentions at Road Intersections	Yesterday
Behavior Classification Algorithms at Intersections and Validation using Naturalistic Data	Yesterday
Learning-based approach for online lane change intention prediction	Yesterday
A Lane Change Detection Approach using Feature Ranking with Maximized Predictive Power	Yesterday
ManeuverRecognitionOOBN_IITS2012	Yesterday
Car that Knows Before You Do- Anticipating Maneuvers via Learning Temporal Driving Models	Yesterday
dynamic bayesian networks. represetation,inference and learning	29/04/2018
probabilistic graphical models	29/04/2018
PATH AND TRAJECTORY PLANNING	29/04/2018
Sequence analysis of glance patterns to predict lane changes on urban arterial roads	29/04/2018
Probabilistic Driving Models and Lane Change Prediction	29/04/2018
Prediction of Pedestrian Trajectories Final Report	29/04/2018
Algorithms for Matching and Predicting Trajectories	29/04/2018
Target Lane Changing Prediction Method for ACC System	29/04/2018
Surround Vehicles Trajectory Analysis with Recurrent Neural Networks	29/04/2018
Multilayer Perceptron	29/04/2018
Predicting Destinations from Partial Trajectories Using Recurrent Neural Network	29/04/2018
Predicting trajectories of golf balls using recurrent neural networks ANTON JANSSON	28/04/2018
Modeling Trajectories with Recurrent Neural Networks	28/04/2018
Self-learning Trajectory Prediction with Recurrent Neural Networks at Intelligent Intersections	28/04/2018
Social LSTM Human Trajectory Prediction in Crowded Spaces	28/04/2018
Predicting Future Lane Changes of Other Highway Vehicles using RNN-based Deep Models	28/04/2018
Classification of Highway Lane Change Behavior to Detect Dangerous Cut-in Maneuvers	28/04/2018
Lane-Change Detection Based on Vehicle-Trajectory Prediction	28/04/2018
A Markov Model for Driver Turn Prediction	28/04/2018