# Prediction of Lane-Changing Maneuvers with Automatic Labeling and Deep Learning

**Vishal Mahajan[1], Christos Katrakazas[2], and Constantinos Antoniou[1]**

SCH Univ.

Dept. of AI and Bigdata

Jaegyun Im

# contents

# 1. Introduction

Recent approaches

- CAVs (Connected and autonomous vehicles)
- AVs (Autonomous vehicles)

Predicting the trajectory of surrounding vehicles → complex calculation

Perfect for real-time applications X

Learning requires a lot of trajectory data

# 1. Introduction

This paper aim

Unsupervised labeling and subsequent prediction of lane-changing maneuvers

HighD Dataset

Density-based clustering

Classifying using SVM models

# 1. Introduction

This paper aim

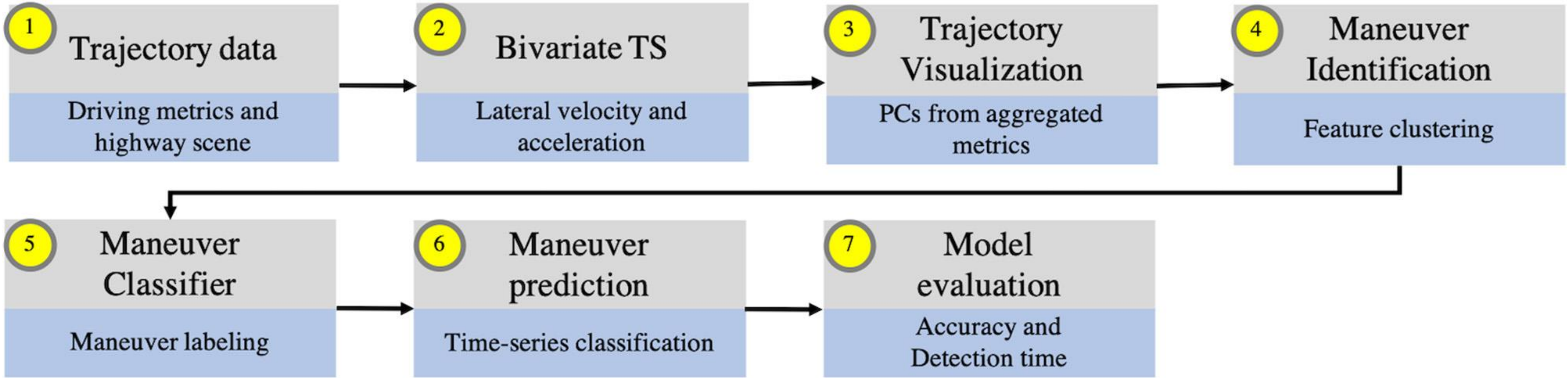
Classifying lane-keeping / lane-changing ( LSTM input )

⇩

Predicting using LSTM model

Improve highway safety through lane change prediction

# 1. Methodology



# 2. Literature Review

There are two types of vehicle maneuvers

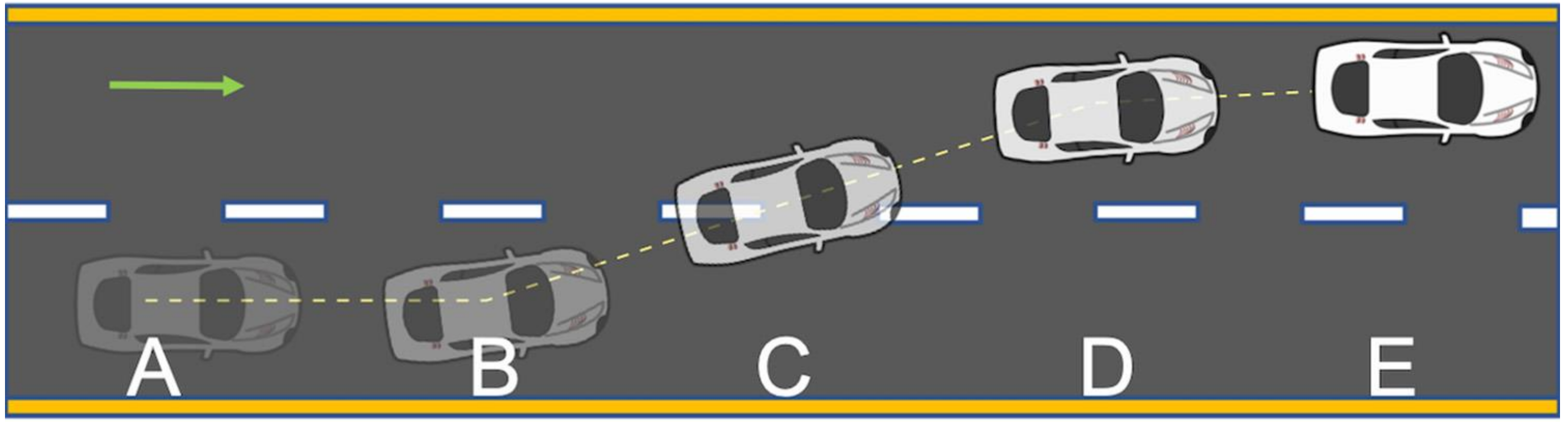
- Lane keeping
- Lane change

Only 44% of lane change engines used turn signals.

Machine learning and data-driven approaches are gaining popularity.

Support vector machines trained with manually labeled datasets show good results in classifying lane keeping and lane changing

# 3. Methodology




It consists of two lane keeping stages (AB and DE) and one lane changing stage (BD)

If all points (B, C, D) are within the observed section of the highway, the lane change is fully implemented

# 3. Methodology

DBSCAN(Density-Based Spatial Clustering of Applications with Noise)

Input :  Vehicle speed and acceleration data

SVM (Support Vector Machine)

Input : Based on clusters formed through DBSCAN

Classifies whether each vehicle trajectory corresponds to lane keeping
or lane changing maneuvers.

LSTM
Input : Bivariate time series data of lateral speed and lateral acceleration of the vehicle.

Input to the LSTM model along with the SVM classification results and used to predict the
vehicle's future lane change behavior

# 3. Methodology

Data preprocessing and feature extraction

$(x_v^t)^n = \{(v_y^t)^n\}$

$(x_a^t)^n = \{(a_y^t)^n\}$

$(x_v^t)^n$, $(x_a^t)^n$ : Indicates the lateral speed $\{(v_y^t)^n\}$ and lateral acceleration $\{(a_y^t)^n\}$ at time t for the n-th vehicle

$(x^t)^n = \{(x_v^t)^n, (x_a^t)^n\} = \{(v_y^t)^n, (a_y^t)^n\}$

$(x^t)^n$ : a vector containing both lateral speed and acceleration at time t of the n-th vehicle

# 3. Methodology

Calculate statistical features

$\mu_i^n = \frac{1}{N^n} \sum_{t=t_b}^{t=t_e} (x_i^t)^n, \forall i \in \{ v, a \}$ : Average for the n-th vehicle i-th feature

$\sigma_i^n = \sqrt{\frac{1}{N^n} \sum_{t=t_b}^{t=t_e} ((x_i^t)^n - \mu_i^n )^2}$ : Standard deviation for the i-th feature of vehicle n

PCA

$P_1^n, P_2^n = f(\sigma_1^n, \sigma_2^n, \mu_1^n, \mu_2^n)$ : $P_1^n, P_2^n$ are the first and second main components obtained through PCA

$f$ is a function that performs PCA conversion

# 3. Methodology

SVM classification

$$\left(M_j^t\right)^n = h((x^t)^n),$$

$(M_j^t)^n$ is a maneuver class label at time t of the n-th vehicle

$h$ is an SVM classifier

$(x^t)^n$ is a feature vector at time t of the nth vehicle

# 3. Methodology

Time series data conversion

$$[X, y]^n =$$

$$\Big[\underbrace{\left(x^{t_b}, x^{t_b + f}, x^{t_b + 2f} \ldots x^{t_b + N-2f}, x^{t_b + N-f}\right)^n}_{\text{Sampled time series data}}, \underbrace{\left(M^{t_b}, M^{t_b + f}, M^{t_b + 2f} \ldots M^{t_b + N-2f}, M^{t_b + N-f}\right)^n}_{\text{Manuvers class label at that time}}\Big]$$

Sampled time series data

Manuvers class label at that time

$b$ : Buffer length (fixed to 10)

$k$ : Length of historical data usage

$p$ : Length of prediction horizon

$f$ : Frame spacing (fixed to 1)

Time Series Data Format Based on Move Pane

$$[X, y]^n = \left[\left(x^{t-kf}, x^{t-(k-1)f} \ldots\ldots x^{t-2f}, x^{t-f}, x^{t}\right)^n, \left(M^{t + pf}\right)^n\right]$$

# 3. Methodology

## LSTM Networks

### LSTM Layer Settings

- It contains two LSTM layers, each of which has 50 units.
- Layers are structured on top of each other → enabling higher temporal dependency learning
- Dense Layer: The second LSTM layer is connected to a dense system containing 20 neurons
- Dense layers are FC structures

Output layer:

- Connect to an additional dense layer (a layer with 20 and 10 neurons)
- Final layer is a dense layer with softmax activation function
- Optimizer : Adam
- Using Dropout

# 3. Methodology

Evaluation Criteria

DBSCAN

- Evaluation index used: Silhouette score

- Measure the quality of clustering by comparing the distance between data points in a cluster and the distance between data points in other clusters

- Score range: -1 to 1

- The closer to 1, the better

# 3. Methodology

Evaluation Criteria

SVM

- Evaluation Indicators Used: Precision, Reproducibility, Accuracy

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Acuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

# 3. Methodology

Evaluation Criteria

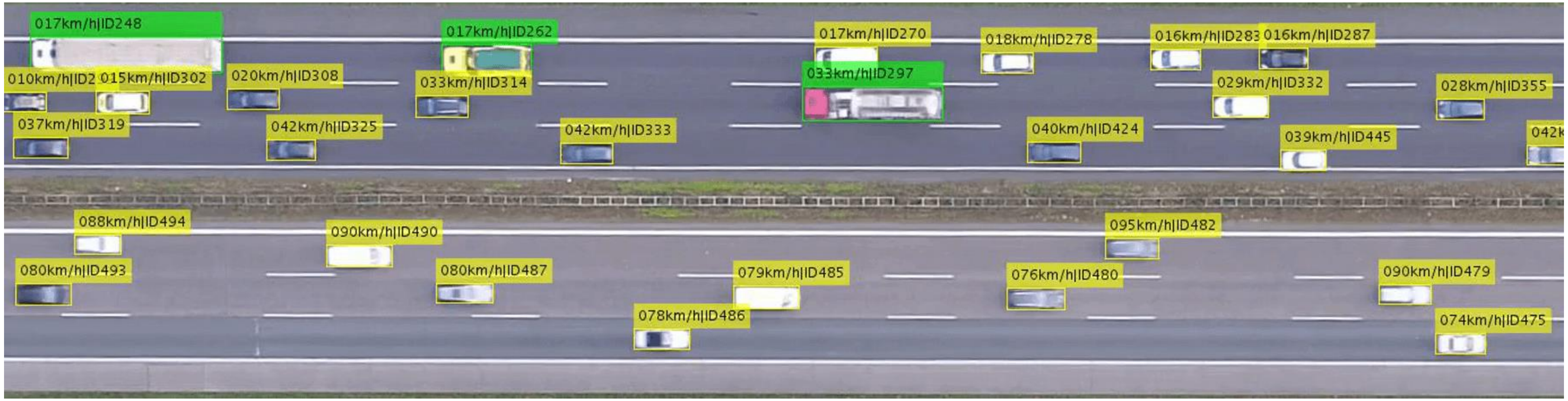
LSTM

- Accuracy and loss, Advance Detection Time (ADT)

- ADT represents the average time the model predicts a lane change before the vehicle crosses the lane mark

$$\mathrm{CE} = -\sum_{i}^{M} M_i \log(s_i)$$

# 4. Data Description



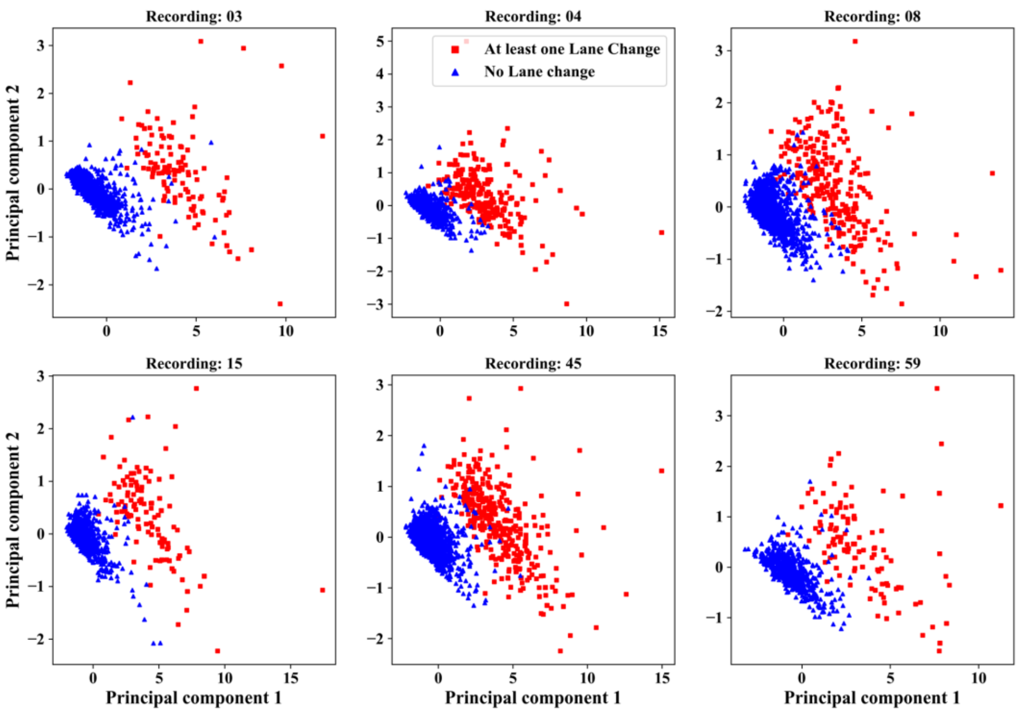

Using HighD Trajectory Datasets

60 greenings, 16.5 hours, 6 locations, 25 Hz frame frequency, 4- and 6-lane highways
45,000km total, 5600 complete lane changes

Using lateral speed and lateral acceleration time series data, extracting training data from specific recorded data only

# 5. Result

## PCA Results



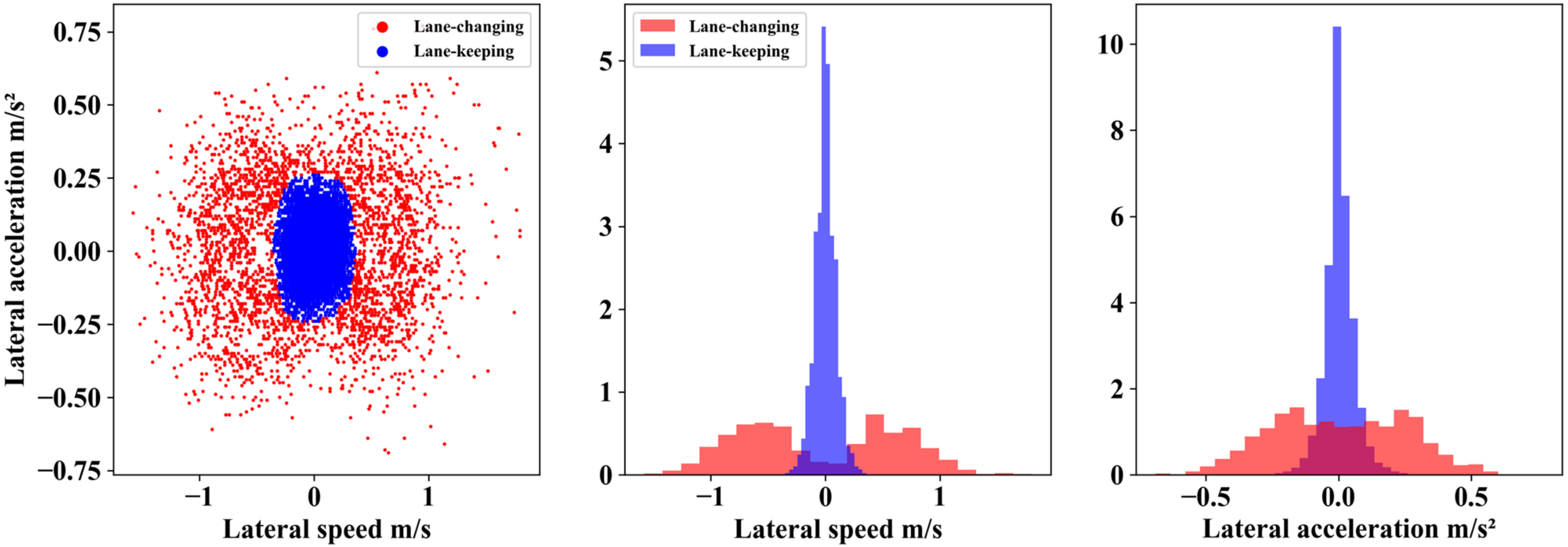

$$P_1^n, P_2^n = f(\sigma_1^n, \sigma_2^n, \mu_1^n, \mu_2^n)$$

$\sigma_1^n, \sigma_2^n, \mu_1^n, \mu_2^n$ account for 98% of the variance

$P_1^n$ accounts for up to 94% of the variance

Show that lateral speed and lateral acceleration are useful in distinguishing lane change from lane keeping

# 5. Result




Results of density-based clustering

Eps : 0.05

Minimum sample 80

Euclidean Metric

The silhouette score is 0.74

# 5. Result

SVM

| Maneuver class | Precision | Recall |
|---|---|---|
| Lane changing | 0.99 | 0.99 |
| Lane keeping | 1.00 | 1.00 |

Optimal parameters are C=0.5 kernels that use radial basis functions

The numbers of TP, FN, TN, and FP are 6121, 7, 4442, and 75 respectively

The reproduction rate is 0.99

The precision is 0.98

Percentage of false alarms (1.66%)

# 5. Result

## LSTM

| | | Accuracy (%) | |
|---|---|---|---|
| Look back time (s) | Prediction horizon (s) | RF | LSTM |
| I | 0.5 | 97.2 | 98.8 |
| I | I | 94.5 | 97.6 |
| I | 2 | 88 | 93.0 |
| I | 3 | 83 | 88 |

Look back time is judged not to affect the results of the model

RF model is fixed with number of estimation = 10, Depth of the tree = 15

Accuracy is significantly lower for prediction times greater than 1 second.

An lstm model with a prediction horizon of 0.5 seconds with high accuracy is used for the test data

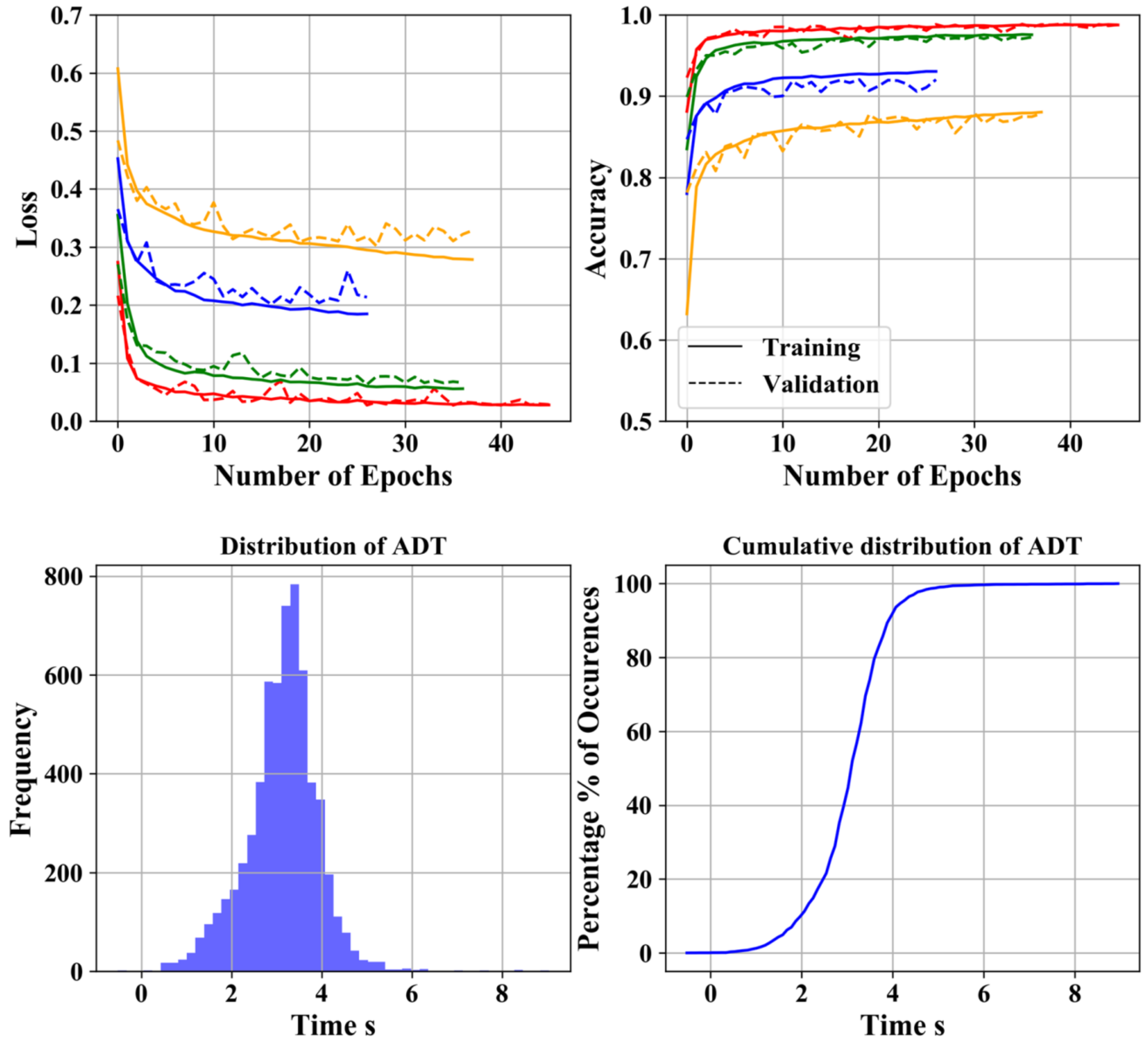
The LSTM model can predict lane change maneuvers with high accuracy 0.5 seconds before start-up

# 5. Result

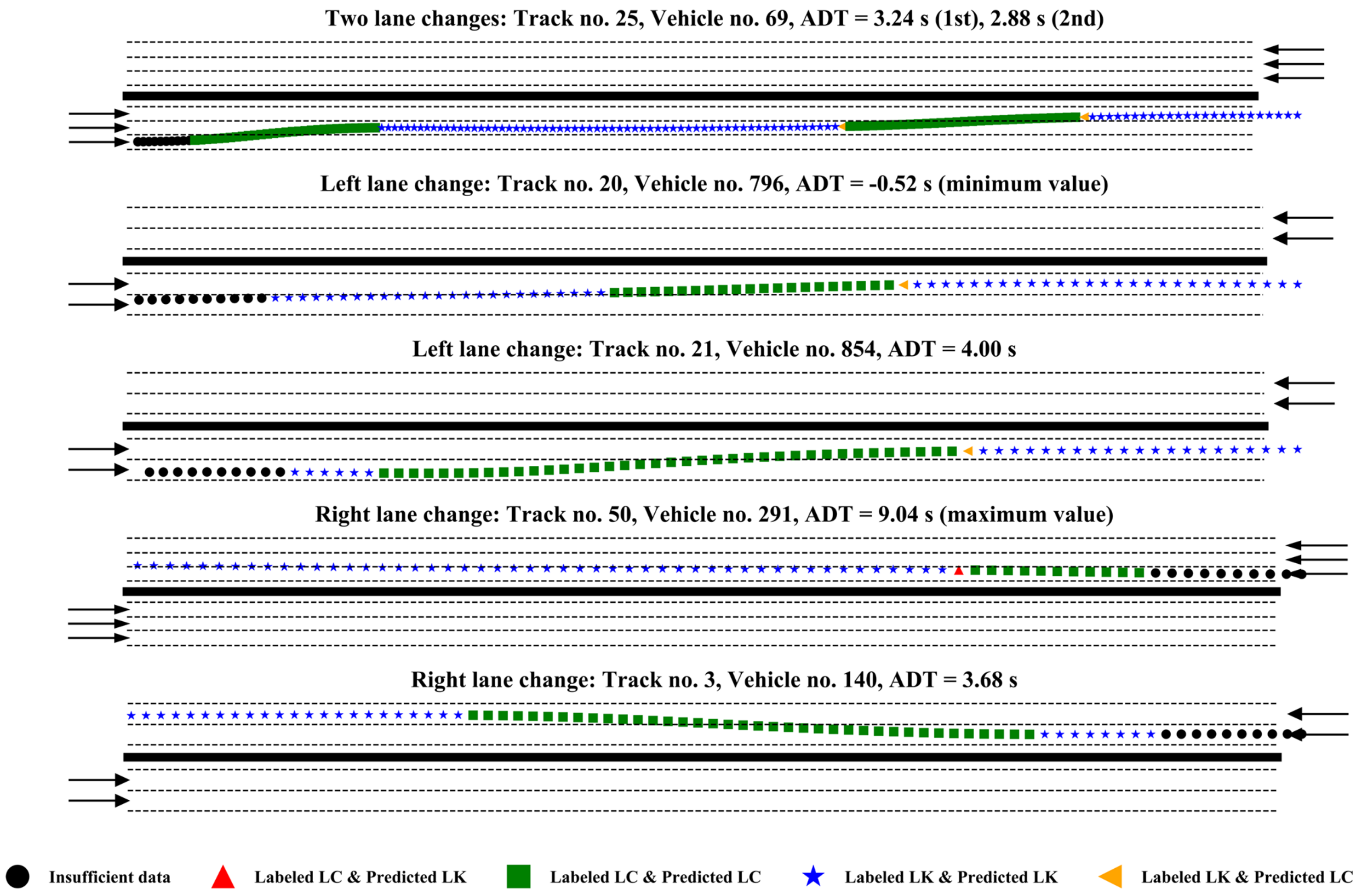
LSTM

ADT average: 3.18 seconds

Standard deviation 0.98 seconds



# 5. Result




LSTM models can detect lane changes with high probability at least 3 seconds before the vehicle crosses the lane mark

# 6. How to apply

- Create a full straight road like a HighD dataset

- Using the clustering technique(Consider)

- Using machine learning or deep learning technology to the lane keeping and classify the lane change maneuver (Consider)