Driver Behavioral Learning
Modeling Distance to Preceding Target Vehicle

Akshay Rakheja
Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario

Abstract

Data from Video Logging Tool (VLT) is used to predict the distance the driver maintains between the subject vehicle and a target vehicle using a vanilla LSTM model. Features such as Average speed, Inverse Time to collision(TTC), longitudinal and polar distances from the preceding vehicle are used for the regression problem that predicts TTC.

Introduction

The goal of this research project is to model the behavior of the driver in the subject vehicle using a myriad of data sources including GPS, Video Logging Tool (VLT) and CAN bus.

Of the several behaviors a driver exhibits, maintaining the distance to the preceding target vehicle is of great importance. This distance can be perceived in different ways using features like Longitudinal and Polar distances from the preceding target vehicle and TTC. The Longitudinal and Polar distances measure the distance between the center of subject vehicle's front bumper and the center of target vehicle's rear bumper. Inverse Time to Collision (TTC) is calculated by the VLT using the above mentioned distances and the relative velocity between the two cars.

$$TTC = \frac{RelativeVelocity}{PolarDistance}$$

TTC represents how close the driver of the subject vehicle follows the preceding target vehicle in terms of seconds and is therefore a feature of interest here. TTC is thus predicted here by modeling it as a regression problem.

Data Preprocessing

Before we begin preprocessing data, we split our drives into training and testing sets. 75% of the data is used for training while the remaining 25% is used for testing. Data collected from VLT comprises of over 700 features for the 12 objects it can detect at once. Most of them are not relevant to the problem at hand (predicting TTC values). The data set also consists of drive segments that belong to country roads, city streets and intersections. To keep our analysis simple, we decided to pick only the highway segments from the route of our subject vehicle. Since our goal is to model the driving distance between the subject vehicle and target vehicle in terms of TTC, we only need to consider the vehicle in the same lane as the subject vehicle when feeding the data into the model. While going over the data, it became clear that not all timesteps incremented evenly and hence, were not continuous. To accommodate for the lack of an independent variable, Image Indices is used. The data set also needs to be scaled to suppress the effect of outliers. Also, the data needs to be in the correct shape to be fed into the model. Hence, it needs to be preprocessed before it can be fed into the model. The steps involved are as follows:

- Extract the necessary features
- Extract relevant highway segments from the drive
- Extract features only relevant to the preceding target vehicle
- Merge the data points by their Image Indices (used as independent variable)
- Scale data
- Remove discontinuities between different drive segments

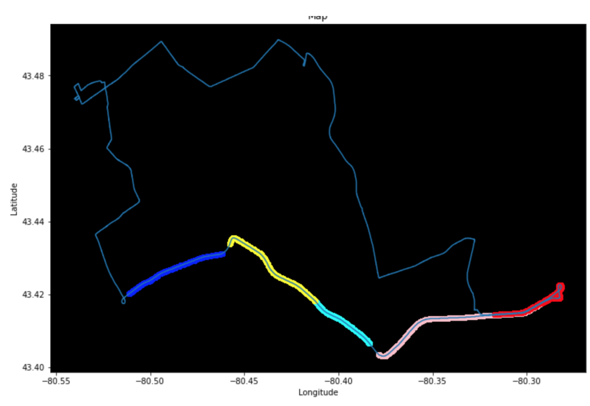

Figure 1: Highlighted Drive Segments within a route

Using the coordinates from the GPS data, Figure1 shows the different drive segments(highlighted) the subject vehicle follows.

The highlighted segments in Figures 2 and 3 show the Training and Testing data points for two different drives respectively.

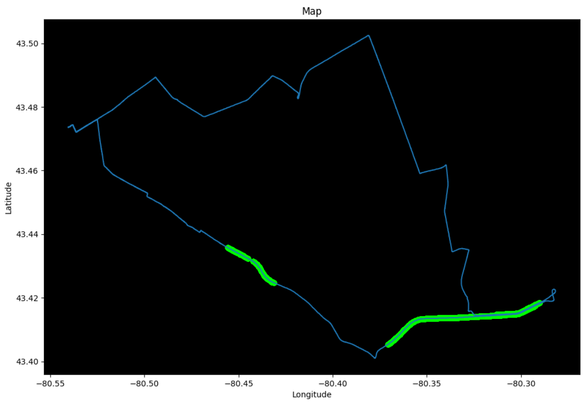

Figure 2: Training Segment highlighted for a drive

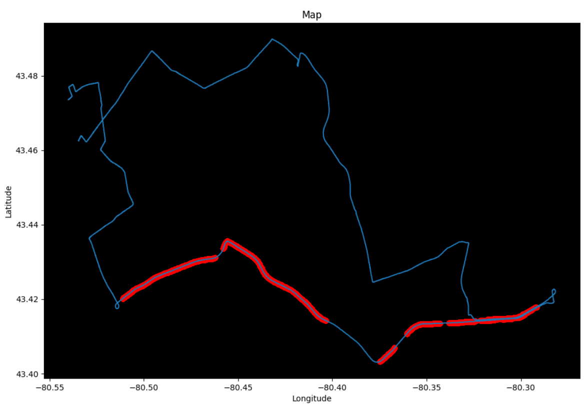

Figure 3: Testing Segment highlighted for a drive

Preprocessed data filters down the data to 10 meaningful features that will help us predict the TTC values. The features used are as follows:

- Relative Velocity between the subject and target vehicles
- Longitudinal Distance between the subject and target vehicles
- Polar Distance between the subject and target vehicles
- Inverse Time to Collision readings for the target vehicle relative to the subject vehicle
- Average speed of the subject vehicle
- Turn Signal on the target vehicle
- Brake Lights on the target vehicle
- Confidence level that VLT attributes to the target vehicle
- Segment number of the higway the subject vehicle is on
- A binary flag indicating the presence of a target vehicle

With the necessary features, data is passed through a function that creates X_train and Y_train data structures. To create these data structures we decided to look at past 5 seconds of data (200 Image Indices) to predict 1 second into the future (40 Image Indices). This results in X_train as a 3d tensor with shape (n_samples - past_frames- future_frames, past_frames, number of features) and y train as a 2d tensor with shape (n_samples, future_frames).
Similarly, using the preprocessing steps above, X_test and y_test are created.

Model

A vanilla LSTM model is used for this regression problem at hand. A sequential model with a couple of LSTM layers is used. Each followed by a L2 regularization factor of 0.01 to avoid overfitting. Both the LSTM layers consist of 10 units each (depicting the dimensionality of the output space). The 2 LSTM layers are also followed by a fully connected layer with an output dimensionality of 40. The output dimensionality is kept 40 to predict 1 second (40 Image Indices) of TTC values given an input of 5 seconds (200 frames or Image Indices).
The model also uses an Adam optimizer along with Mean Squared Error Loss function.

Training

The model described above is run for 50 epochs with a batch size of 256. It is optimized for having a low validation loss using callbacks such as EarlyStopping, ReduceLROnPlateau with a patience of 10.

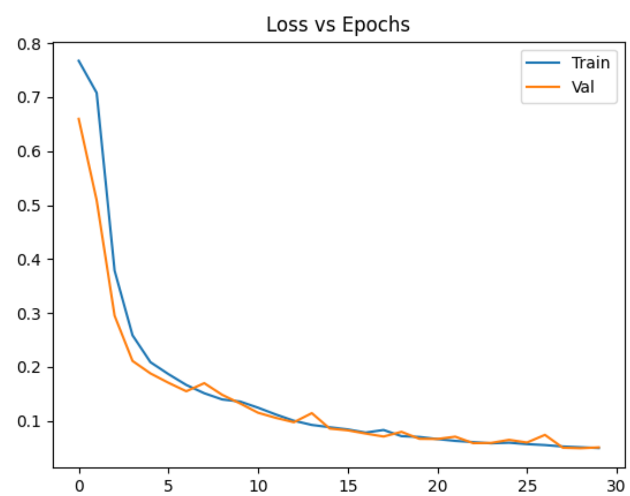

Figure 4: Loss vs Epochs

Results

To gauge how the model performs, a graph showing actual TTC values is plotted against the predicted values. This is done for 10 random samples representing 5 seconds (200 frames) of data followed by 1 second of output (40 frames). For a better understanding of the results, a similar graph as above is plotted representing an entire minute of data (2400 frames) along with 55 seconds of predictions (2200 frames). Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are calculated for the predicted TTC values.

This model provides baseline results that we can further improve upon by swapping the model. For example, we can use LSTM with attention or Transformers to attain better TTC values and still not overfit.

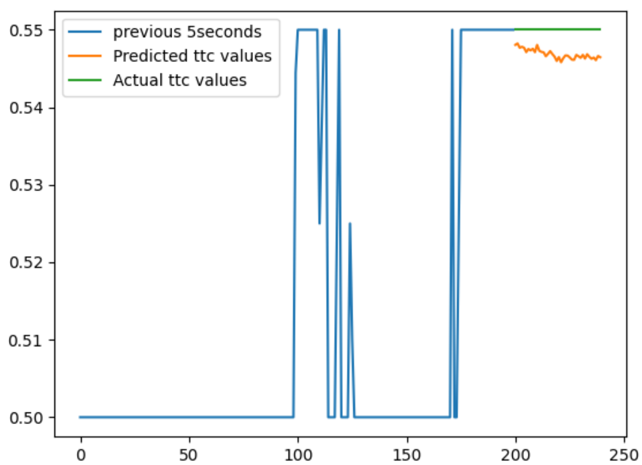

Figure 5: Actual vs Predicted TTC values for random sample: 1, RMSE: 0.002149388520205403, MAE: 0.00164124208152861

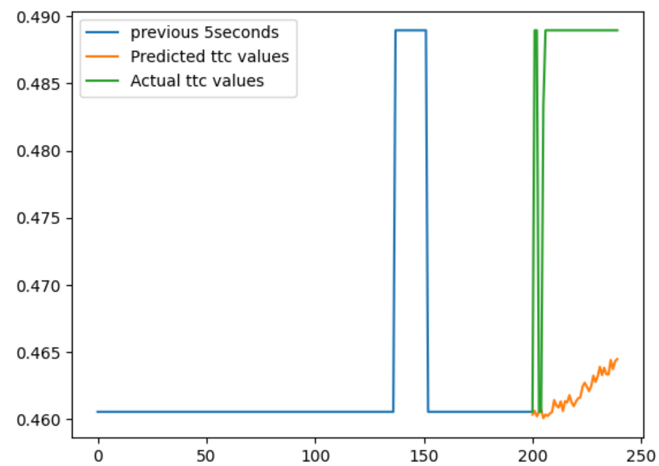

Figure 6: Actual vs Predicted TTC values for random sample: 2, RMSE: 0.003864071359229251, MAE: 0.003841979163184783

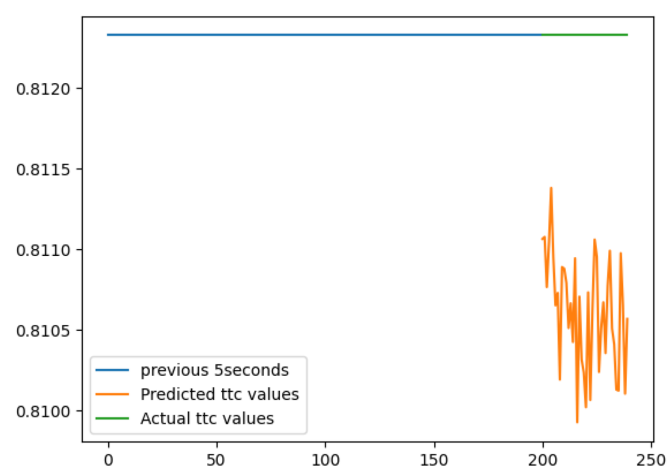

Figure 7: Actual vs Predicted TTC values for random sample: 3, RMSE: 0.032152850932150125, MAE: 0.026379011323054626

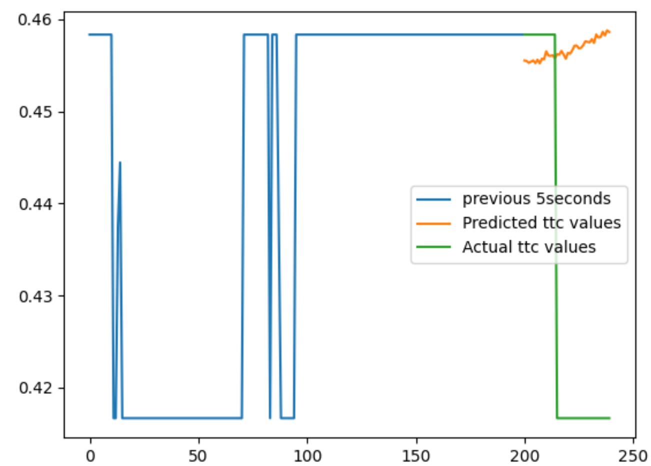

Figure 8: Actual vs Predicted TTC values for random sample: 4, RMSE: 0.019688818036188264, MAE: 0.009230404794216097

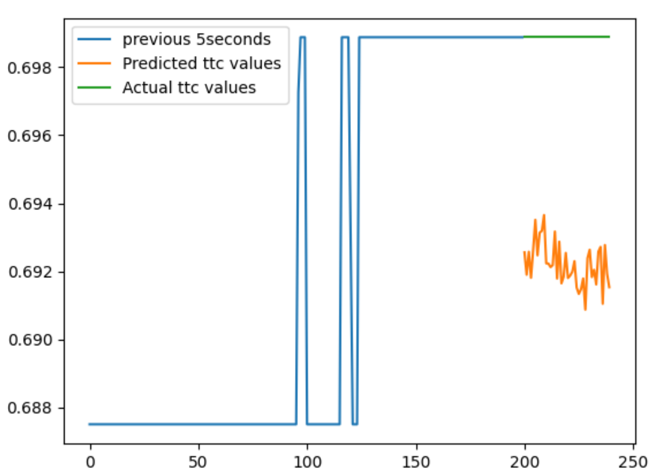

Figure 9: Actual vs Predicted TTC values for random sample: 5, RMSE: 0.006710424744166728, MAE: 0.006680741001622858 6

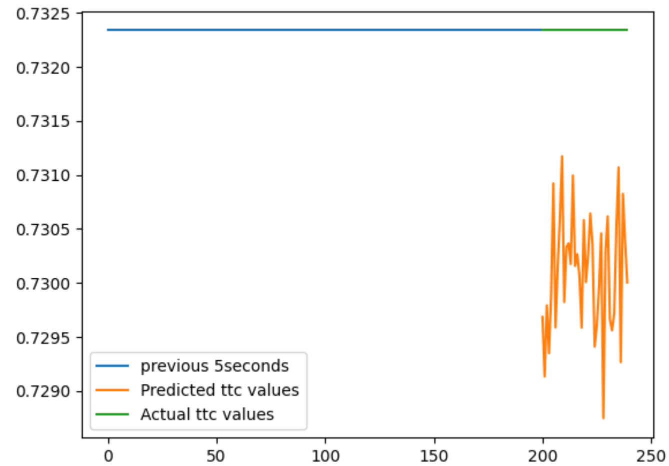

Figure 10: Actual vs Predicted TTC values for random sample: 6, RMSE: 0.025783275908015567, MAE: 0.024772503350153293

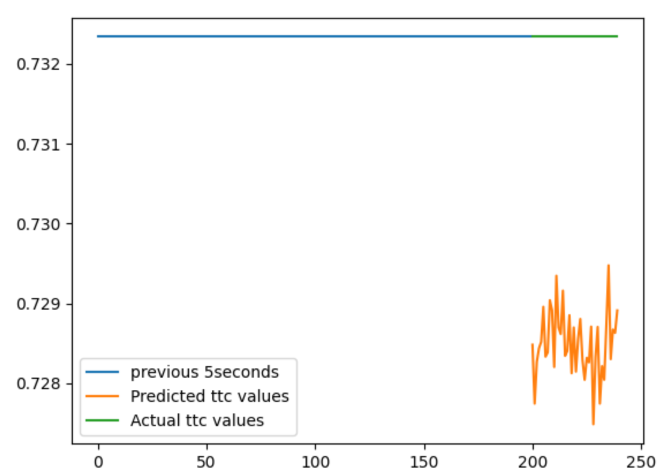

Figure 11: Actual vs Predicted TTC values for random sample: 7, RMSE: 0.025079847048686817, MAE: 0.022593399442923 44

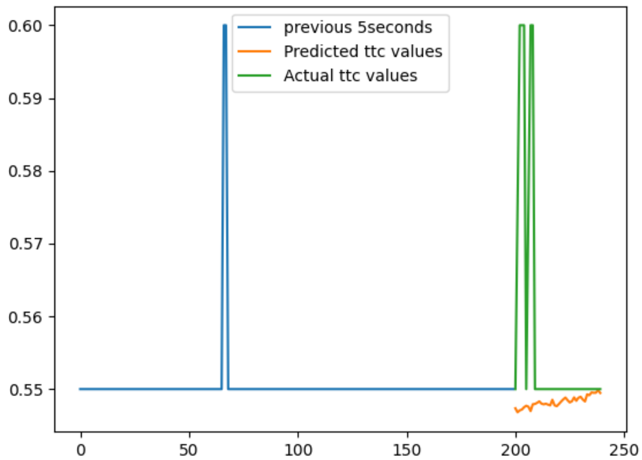

Figure 12: Actual vs Predicted TTC values for random sample: 8, RMSE: 0.0017456104219638082, MAE: 0.00171053026412471

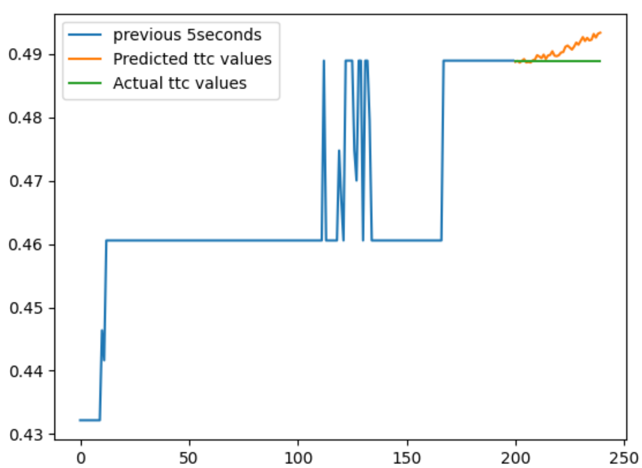

Figure 13: Actual vs Predicted TTC values for random sample: 9, RMSE: 0.0023055730956544466, MAE: 0.0022388675384670707

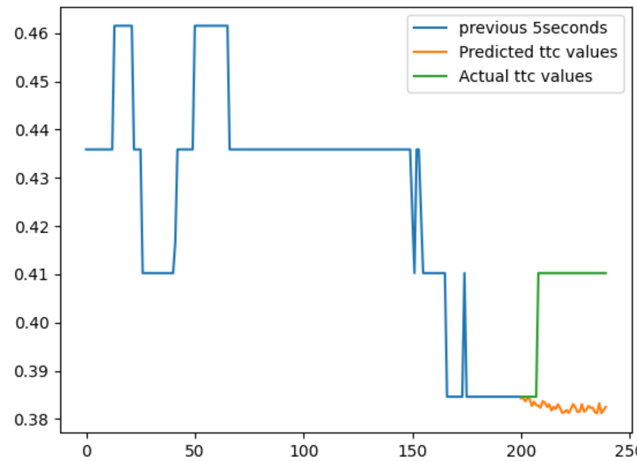

Figure 14: Actual vs Predicted TTC values for random sample: 10, RMSE: 0.0031971453910544768, MAE: 0.0031391754746436405

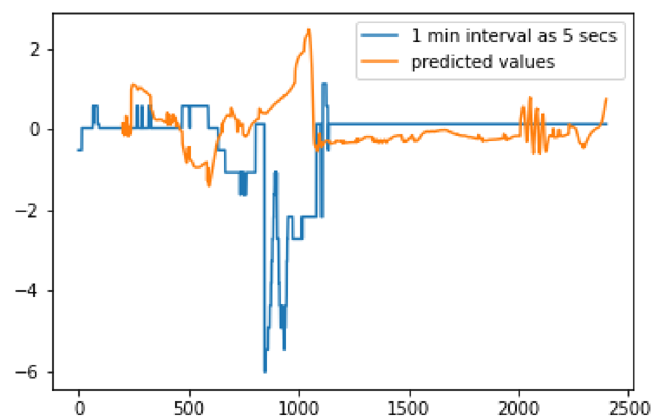

Figure 15: Actual vs Predicted TTC values for 1 minute of input data