Data-Driven Car Following Model

DAB402 – Capstone Project

Vineet Dhamija
Department of Data Analytics for Business
St Clair College, Windsor, Ontario, Canada. N9A 5K3
Email: vd17@myscc.ca

Neel Chaudhari
Department of Data Analytics for Business
St Clair College, Windsor, Ontario, Canada. N9A 5K3
Email: nc57@myscc.ca

Rakesh Singh
Department of Data Analytics for Business
St Clair College, Windsor, Ontario, Canada. N9A 5K3
Email: rs334@myscc.ca

Supervised by
Dr. Umair Durrani
St. Clair College

Abstract

The very first prototype of car following model, GM Model was put forward in the year 1958, it was based on the assumption that the acceleration was proportional to the relative speed. This model can be termed as an assumption, which is the basis of the models widely in use today, such as Intelligent driver model(IDM). In current age of Big Data with advancement in Data Science we can perform the same task better, basing our predictions on real world data and Driver behaviour. With advanced technology, people were able to generate data using drone videography which can be used to create a data-driven car following model. We have used one such available Next Generation Simulation (NGSIM) dataset and trained three models Random Forest, K-Nearest Neighbours(KNN) and Convolution Neural Network(CNN) to predict acceleration and calculate rest vehicle trajectory of a Subject (human) vehicle following any autonomous or Human Driven Vehicle in a single lane car following scenario. Further comparison has been made between the predicted trajectories and the validate them with the errors of the predictions.

Table of Contents

List of Figures

List of Tables

Introduction

According to a report released by the research house Bernstein, the number of cars on the road is projected to reach two billion by 2040, especially in developing countries like China and India due to the rise in the population as well as an increase in GDP[1]. And 25% of the collisions in Canada is rear-end collision. This problem calls for a more generic and realistic solution. Data-driven car following model can be used to solve this problem[2].

Many organizations, educational institutions, and individuals are working on different car-following models and many models are already present for use, but each model has its own limitations and can not be used in every situation as the driving condition changes in every part of the world. Most of the car following models available are assumption-based and not data-driven which has the problem of biasness, hence data-driven models when used in any simulation or in any automotive vehicle will give more accurate results compared to assumption-based models. These models can be used in Forward collision warning systems, Simulations for road construction, and planning to name a few.

Problem Statement and Research Question

Every year many papers are written on car following models which extensively talk about velocity prediction but do not talk about acceleration prediction. Very few of the papers discuss acceleration predictions and claim to have good RMSE but do not actually show the acceleration trajectory with respect to the actual values or show the RMSE value to do a fair comparison.
In our project, we have created Random Forest, KNN, and CNN models using NGSIM dataset and predict the acceleration of the vehicle and also show both RMSE as well as trajectory plots.

Literature Review

All the Car Following models that have been made falls under either of the two categories, Data-driven or Assumption/Theory based car-following model with most of the models made on Theory based, and very few are Data-driven models. These models can be further classified as Macroscopic, Microscopic, or Mesoscopic models.

Assumption/Theory based Car following model:

Following is the list of basic assumptions on which car following models are prepared:

- The bumper-to-bumper distance between the subject vehicle and the lead vehicle should not be less than a safe distance.
- Soft Braking under normal condition. In critical condition, deceleration exceeds comfortable value and then comfortable braking once the danger is averted.
- The transition between driving modes like from acceleration to car-following mode should be smooth.
- The model should be as parsimonious as possible. Each model parameter should describe only one aspect of the driving behavior (which is favorable for model calibration). Furthermore, the parameters should correspond to an intuitive interpretation and assume plausible values.

Data-Driven Car Following Model:

Data-Driven Car following model as the name tells is made using data and is not based on any assumptions. These models do not require any prior knowledge as well as any assumptions, rather they are purely created using data. Hence, it requires very good quality data to create a model which is as close to reality as possible and can possibly be used in any kind of driving environment.
Data-driven car following model employs equations of motions during prediction to make an accurate prediction and generate good trajectories[3].

$$v = u + at \qquad (1)$$

$$s = ut + \frac{1}{2}at^2 \qquad (2)$$

$$a = \frac{dv}{dt} \qquad (3)$$

$$J = \frac{da}{dt} \qquad (4)$$

u = initial velocity
v = final velocity
t = time at any moment during motion
a = acceleration
J = Jerk

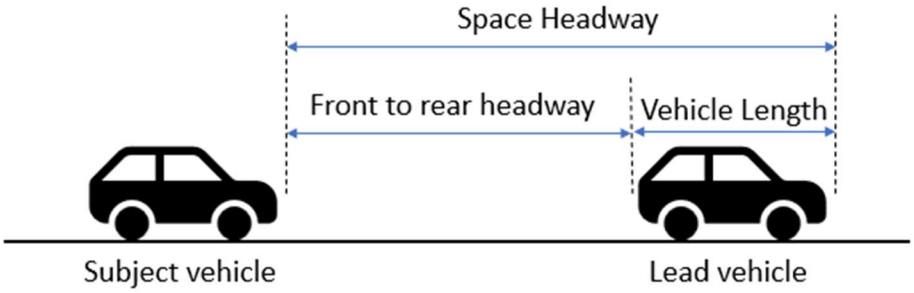

Figure 1. Terminologies often used in car following models

Figure 1 shows the common terminologies which are used in model creation and prediction. These features combined with some other features are used to train the model. While following a vehicle, subject vehicle's driving behaviour is dependent on the lead vehicle. For instance, the spacing between the two vehicle increases as the speed increases as well as based on the type of lead vehicle (more space in case of Heavy vehicle as lead vehicle). Reaction time will change depending on the speed and type of lead vehicle.

Dataset Description

The dataset used in this project is called Next Generation Simulation (NGSIM) and it is publicly available on the U.S. Department of Transportation website. The NGSIM dataset consists of 25 columns and 11.8 million rows of vehicle trajectory data which was captured using a network of synchronized digital video cameras on 4 different locations (US 101, I-80, Peachtree, and Lankershim). The data contains the location of a vehicle at every one-tenth of a second, which gives the exact position of each vehicle relative to other vehicles[4].
A total of 3 different types of vehicle data can be found in this dataset, namely Car, Truck, and Motorcycle. Most of the data were taken from the two freeways i.e., US 101 and I-80, and among the three vehicle types, data on Cars is more as compared to Trucks and Motorcycles. Therefore, we decided to work on only the Freeways. After the Preliminary analysis, we found that some vehicle IDs are present in more than one location meaning that the data from all four locations were taken separately and then merged in a single file. Hence, we separated the data based on location to carry out data cleaning and data transformation and then merge them back.

Dataset link: Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data | Department of Transportation - Data Portal

Data Preparation

Cleaning of data as well as the transformation was done to create the models for better prediction. The Accuracy of the models is directly proportional to the quality of the data used.
As mentioned before, our dataset consists of 4 locations out of which we only took US 101, and I-80 locations data as data from these two locations alone amounts to more than 70% of the data. The Data had many redundant fields which were removed to make the model generic. After following the steps from figure 2, a clean data was obtained.

Before model creation, units of all the fields were changed to a SI unit. In which Vehicle length, local_X, local_Y, and space headway were converted from feet to meters, Velocity was converted to meter/second, and acceleration was converted to meter/second2. After the data was cleaned and the units were changed, L_F pairs (Lead and Following vehicle pair) were created using Preceding and vehicle ID, removed duplicate vehicle pairs present at both locations, and created a new field for space headway from the rear of the lead vehicle and front of the subject vehicle removing dependency on the type of vehicle.
After all the cleaning and transformation following 8 pairs data was available for model creation.

- Car – Car (4742 Pairs)
- Car – Heavy Vehicle (23 Pairs)
- Car – Motorcycle (2 Pairs)
- Free Flow – Car (22 Pairs)
- Heavy Vehicle – Car (22 Pairs)
- Heavy Vehicle – Heavy Vehicle (2 Pairs)
- Heavy Vehicle – Motorcycle (1 Pair)
- Motorcycle – Car (3 Pairs)

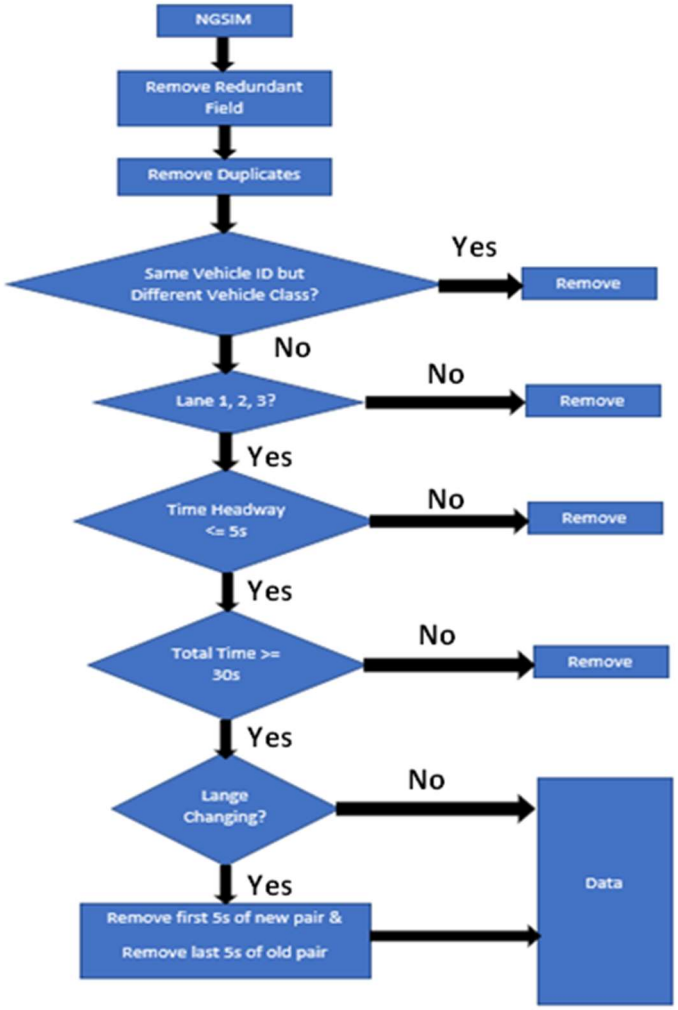

Figure 2. Data Cleaning steps

Methodology

Three models namely Random Forest Regressor, CNN, and KNN were used to create the models to predict velocity and acceleration. To make a fair comparison of all the three models, features, target, and the split of the data were kept the same and their information can be seen in table 1.

Features	Lead Vehicle Rear to Subject Vehicle Front Space Headway, Lead Vehicle Class, Subject Vehicle Class, dv = Subject Vehicle Velocity – Lead Vehicle Velocity, Subject Vehicle Velocity
Target	Acceleration
Train/Validate/Test split	80/10/10

Table 1. Common details of models

The above-mentioned step of shifting the value was common for each model and the other step which was also common in each model is related to the prediction. In the model, the first prediction was done by

taking the very first row value of the data set, and then the output of the prediction was taken and by using the three equations of motions, other values were calculated and fed into a for loop which then did the next prediction. Basically, the output of one prediction was used as the input of the next prediction.

Random Forest:

Random Forest is a supervised ensemble learning method. The ensemble learning method combines predictions from various machine learning algorithms to provide predictions that are more accurate than those from a single model. Random Forest fits several classifying decision trees on different samples of data set and then uses averaging to improve the accuracy and control overfitting. Random Forest can be used as both classifier and regressor, and since our data demands for regressor, we will be using Random Forest Regressor.

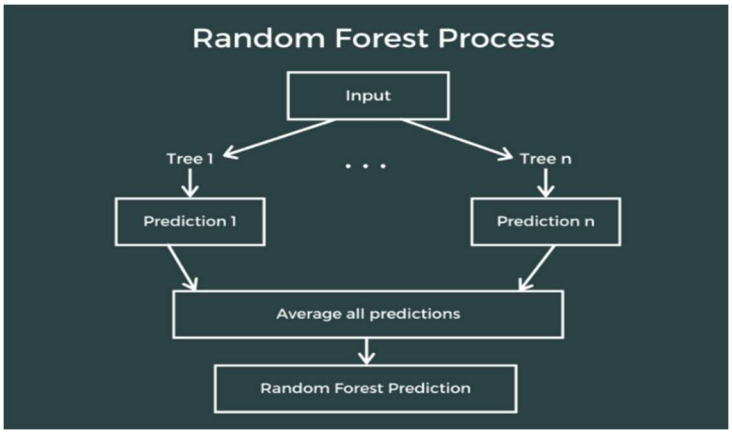

Figure 3. Random Forest Process for n number of trees

Keeping the n-estimator as 30, the model was created for reaction time 0.1, 0.2, 0.3, 0.5, 1, 2, and 4 seconds. Using the predicted values of acceleration, we calculated velocity, spacing, and jerk and plotted the trajectories of predicted values against actual values[5].

Convolutional Neural Network (CNN):

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that can take in an input image, give importance (learnable weights and biases) to various aspects/objects in the image, and be able to distinguish one from the other. In comparison to other classification methods, a ConvNet requires significantly less pre-processing. ConvNet have the capacity to learn these filters and properties, whereas in basic techniques filters are hand-engineered.
Table 2 below gives the complete details of the trained CNN model.

Block	*Layer*	*No. of Nodes*	*Activation Function*
1	Conv1D	16	Sigmoid
	Conv1D	16	Sigmoid
2	Conv1D	32	Elu
	Conv1D	32	Elu
3	Conv1D	32	Tanh
	Conv1D	32	Tanh
	Dense	128	Tanh
	Dense	64	Sigmoid
	Dense	16	Tanh
	Output	1	Elu
Model	Functional		
Epochs	10		
Optimizer	Adam		
Loss Function	Mean Squared Error		
Metrics	Root Mean Squared Error		

Table 2. Trained CNN model

K-Nearest Neighbors (KNN):

The k-nearest neighbors algorithm, sometimes referred to as KNN or k-NN, is a supervised learning classifier that employs proximity to produce classifications or predictions about the grouping of a single data point. Although it can be applied to classification or regression issues, it is commonly employed as a classification algorithm because it relies on the idea that comparable points can be discovered close to one another.

In our case, we adopted 5-neighbors and by means of Euclidean distance, the 5 nearest neighbors were identified.

Result

Below is a summary of the results that has been obtained from all three trained models. We have divided the results in subcategories of the predictions made for Acceleration, Spacing , Velocity and Jerk. Each subcategory has the errors mentioned for various reaction times.
Below is an average of the overall R2 and RMSE of all the models:

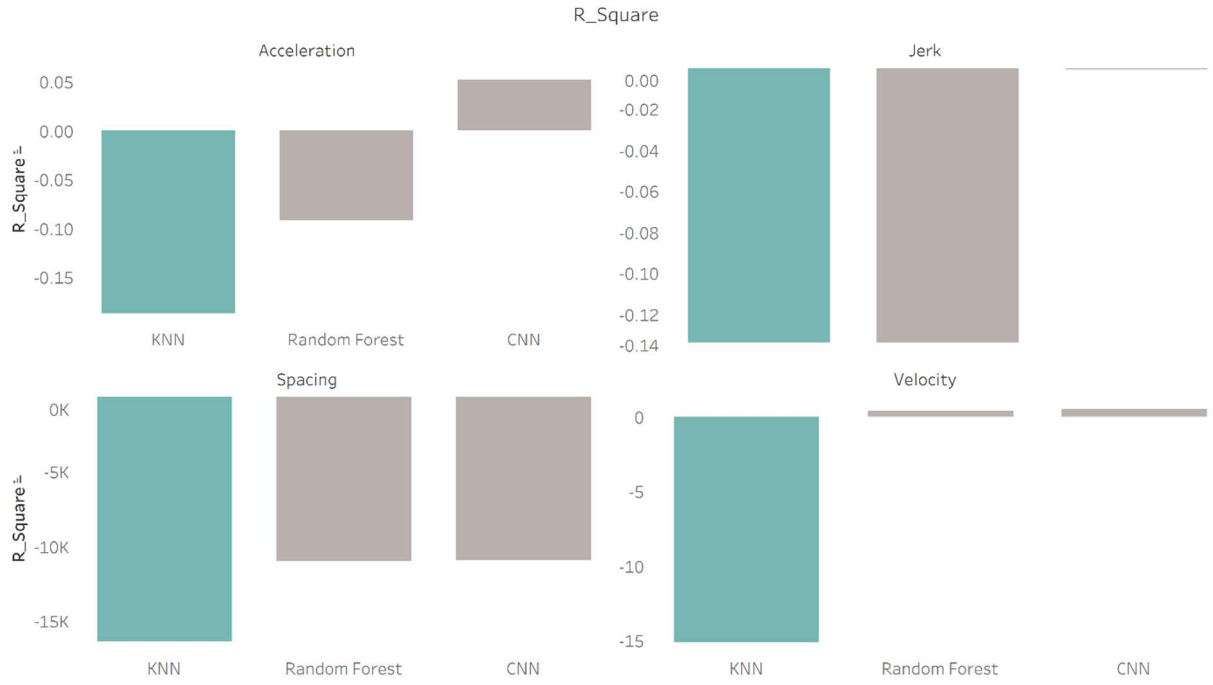

Figure 4. R2 comparison for all the models

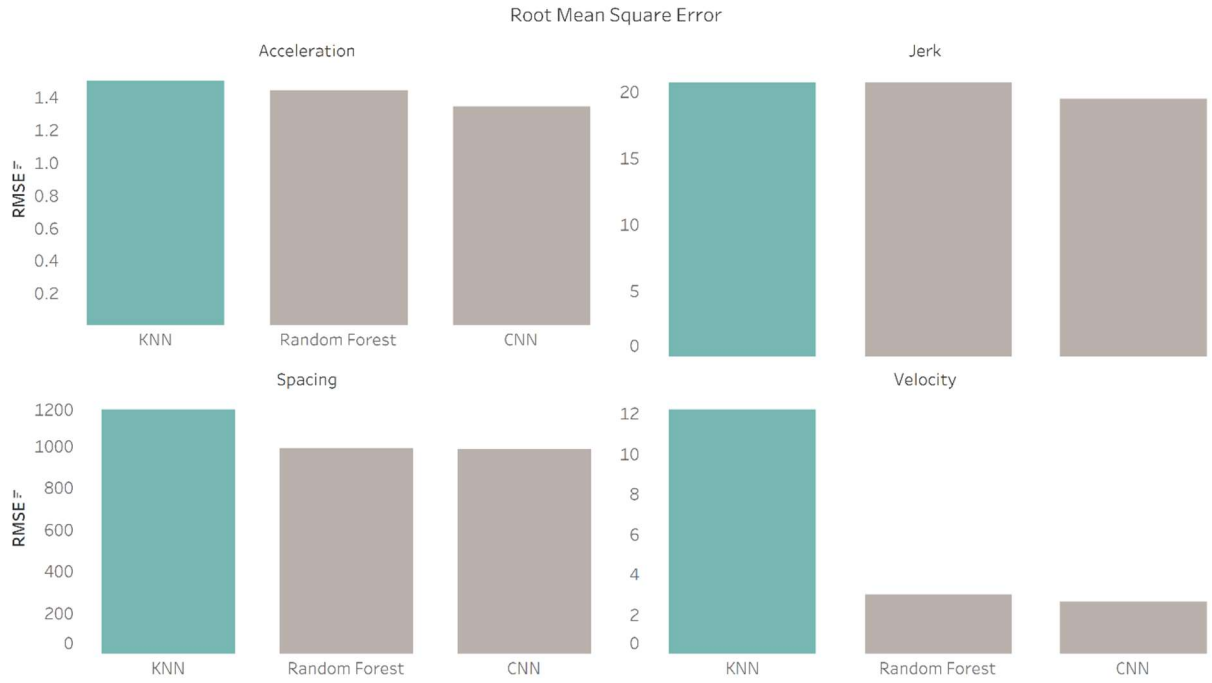

Figure 5. RMSE comparison for all the models

Acceleration:

		Reaction Time (seconds)						
		0.1	0.2	0.3	0.5	1	2	4
RF	R^2	-0.118	-0.113	-0.085	-0.049	-0.056	-0.103	-0.121
	RMSE	1.46	1.45	1.44	1.41	1.41	1.44	1.45
KNN	R^2	-0.121	-0.127	-0.236	-0.131	-0.119	-0.274	-0.314
	RMSE	1.46	1.46	1.53	1.46	1.45	1.55	1.57
CNN	R^2	0.069	0.065	0.071	0.074	0.062	0.028	-0.004
	RMSE	1.33	1.33	1.33	1.33	1.33	1.35	1.37

Table 3. R2 and RMSE for Acceleration Prediction

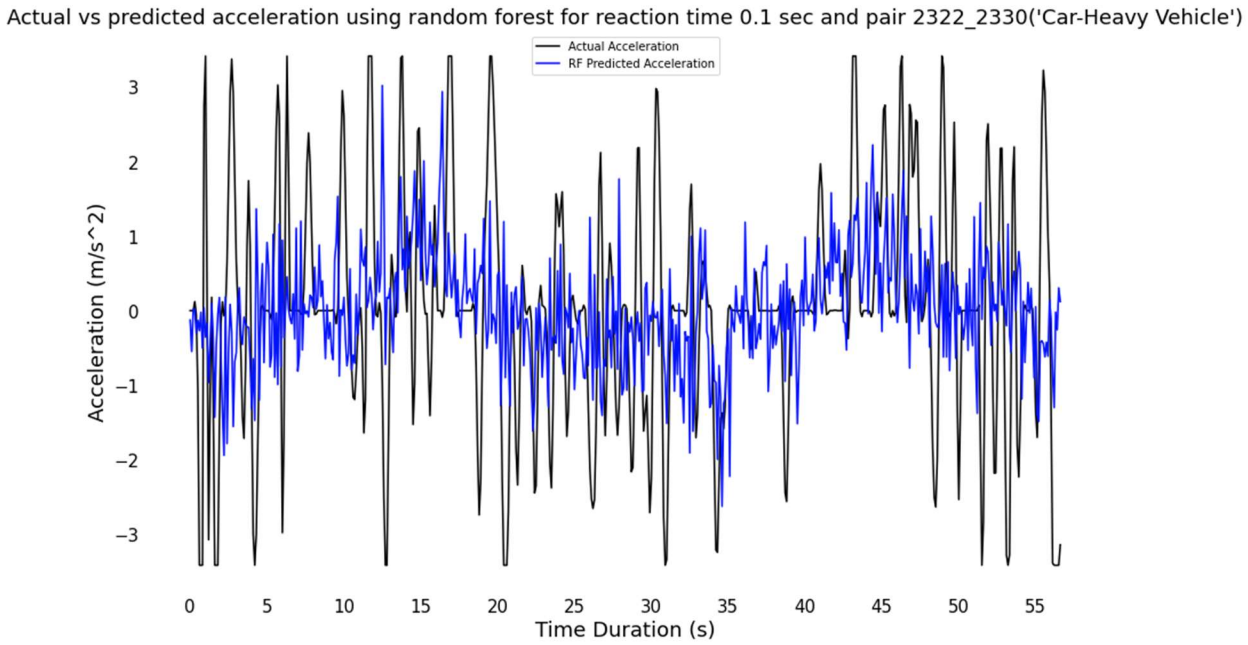

Figure 6. Acceleration for 0.1 reaction time for Random Forest model

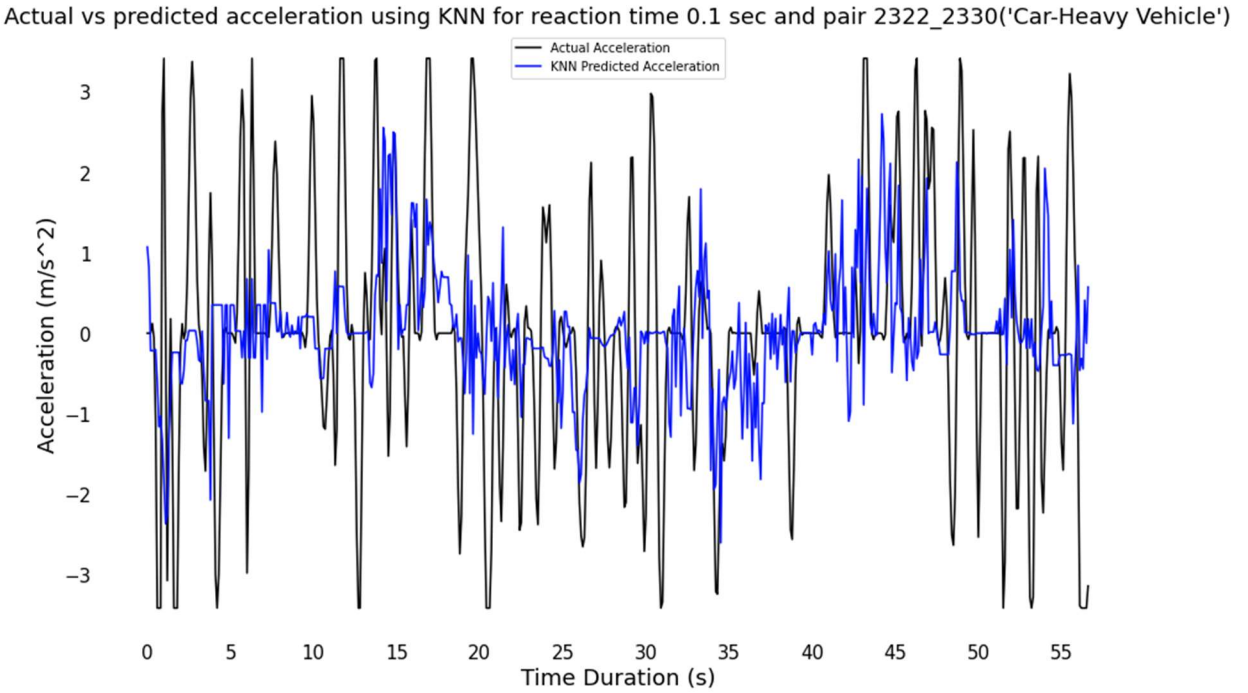

Figure 7. Acceleration for 0.1 reaction time for KNN model

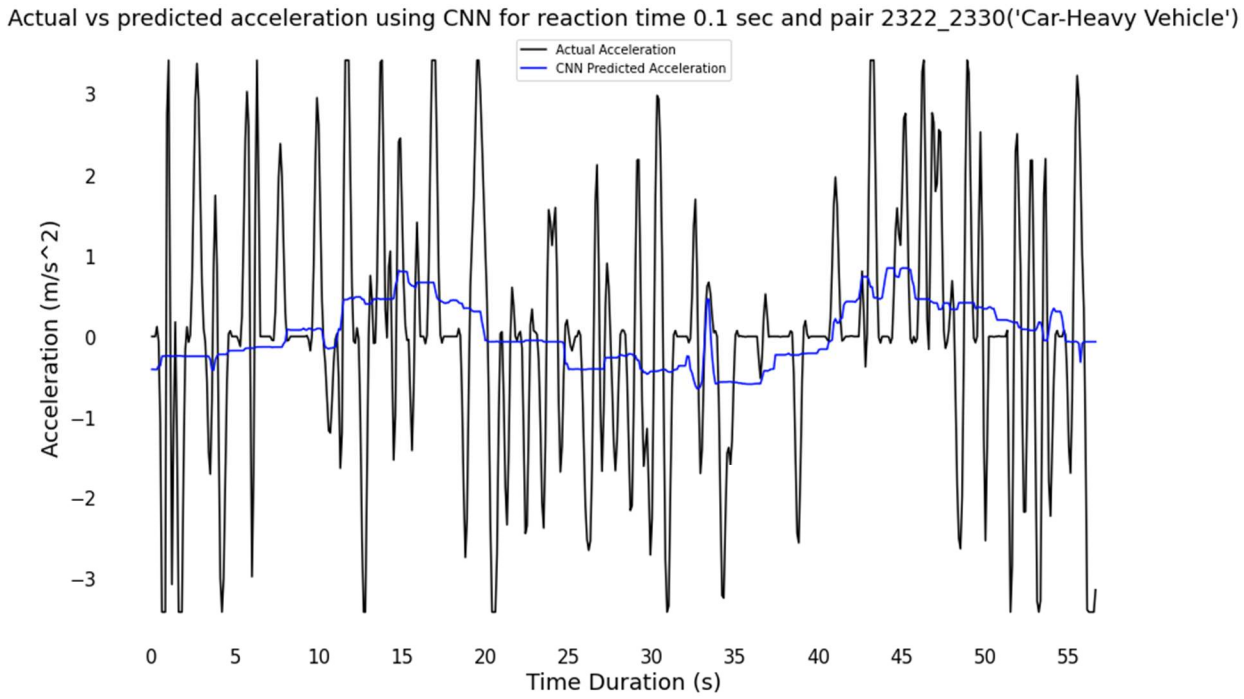

Figure 8. Acceleration for 0.1 reaction time for CNN model

Velocity:

		Reaction Time (seconds)						
		0.1	0.2	0.3	0.5	1	2	4
RF	R^2	0.809	0.837	0.832	-0.091	0.490	-0.021	-0.348
	RMSE	1.73	1.60	1.62	4.14	2.83	4.01	4.62
KNN	R^2	0.702	-3.971	-38.580	-11.957	0.710	0.155	-53.082
	RMSE	2.16	8.82	24.90	14.25	2.13	3.65	29.29
CNN	R^2	0.239	0.820	0.710	0.500	0.601	0.747	0.102
	RMSE	3.45	1.68	2.13	2.80	2.50	2.00	3.78

Table 4. R2 and RMSE for Velocity Prediction

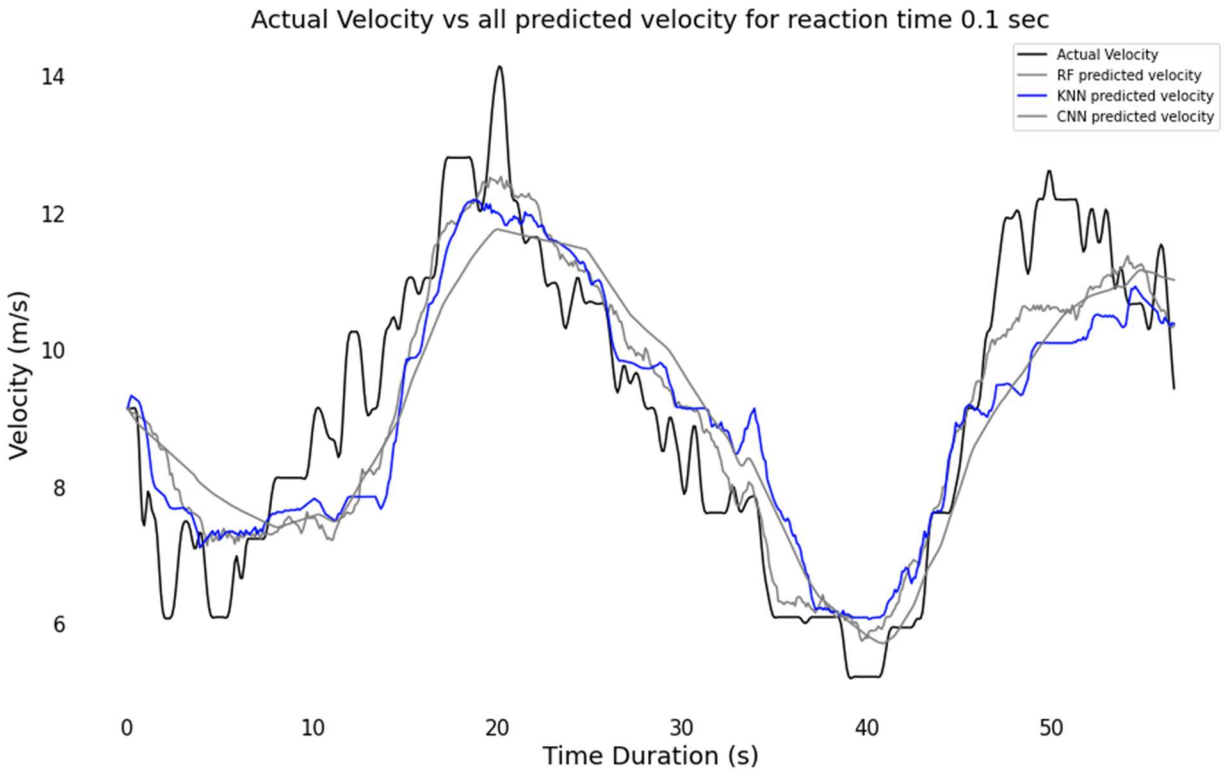

Figure 9. Velocity for 0.1 reaction time

Spacing:

		Reaction Time (seconds)						
		0.1	0.2	0.3	0.5	1	2	4
RF	R^2	-10684.3	-1.636.5	-10627.0	-11323.3	-11065.9	-11193.1	-11452.6
	RMSE	967.24	965.15	964.79	996.05	985.05	991.65	1004.48
KNN	R^2	-9920.7	-12612.5	-23754.8	-14430.9	-11067.9	-10911.9	-32316.3
	RMSE	932.04	1050.97	1442.42	1124.44	985.14	978.99	168.28
CNN	R^2	-11137.4	-10157.1	-10896.3	-11036.3	-10998.5	-10937.9	-11438.0
	RMSE	987.54	943.15	976.94	983.34	982.05	980.15	1003.84

Table 5. R2 and RMSE for Spacing Prediction

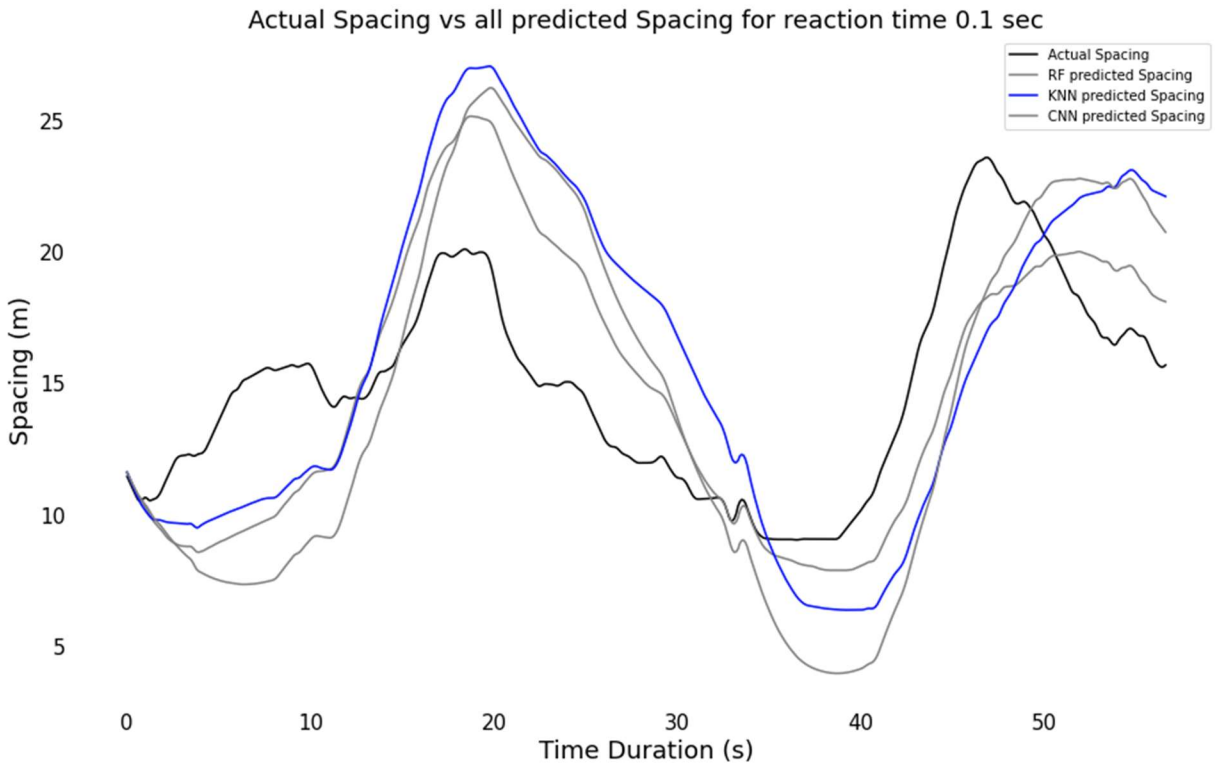

Figure 10. Spacing for 0.1 reaction time

Jerk:

		Reaction Time (seconds)						
		0.1	0.2	0.3	0.5	1	2	4
Random	R^2	-0.172	-0.175	-0.150	-0.114	-0.098	-0.114	-0.115
Forest	RMSE	21.03	21.05	20.82	20.49	20.32	20.42	20.38
KNN	R^2	-0.164	-0.163	-0.150	-0.141	-0.133	-0.119	-0.068
	RMSE	20.96	20.95	20.83	20.73	20.64	20.47	19.95
CNN	R^2	-0.0006	-0.0024	-0.0023	-0.0009	-0.0002	0.0013	0.0000
	RMSE	19.43	19.45	19.44	19.42	19.40	19.34	19.30

Table 6. R2 and RMSE for Jerk Prediction

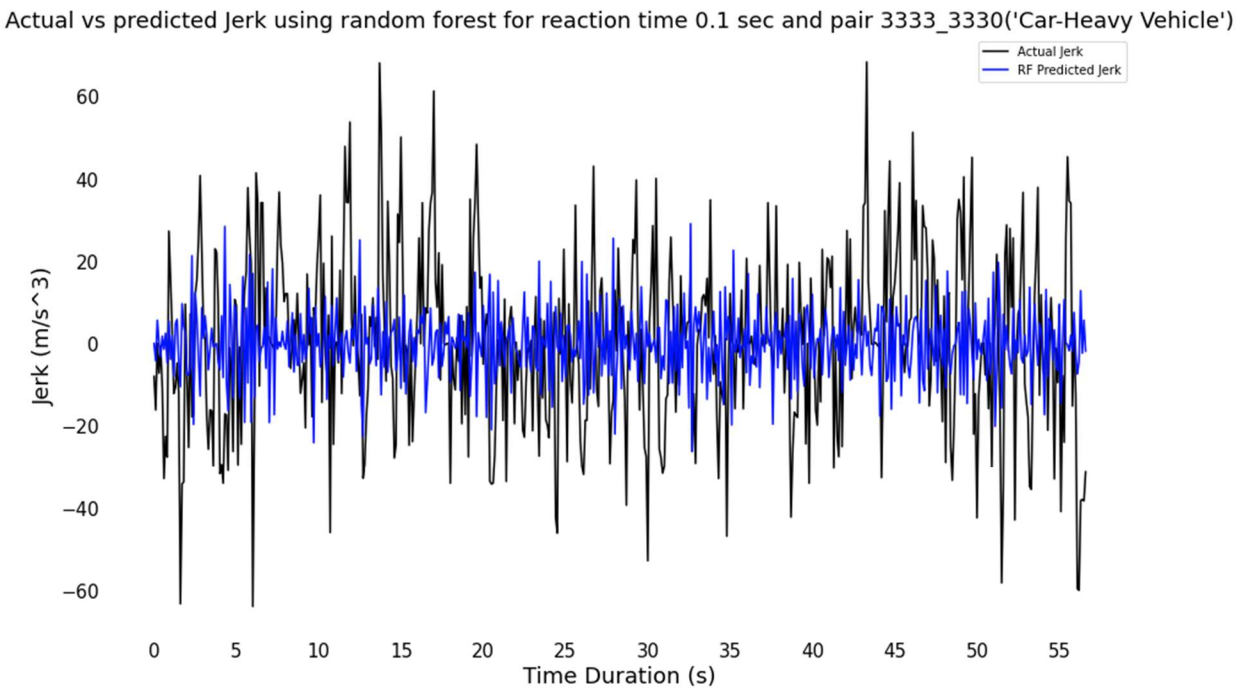

Figure 11. Jerk for 0.1 reaction time for Random Forest model

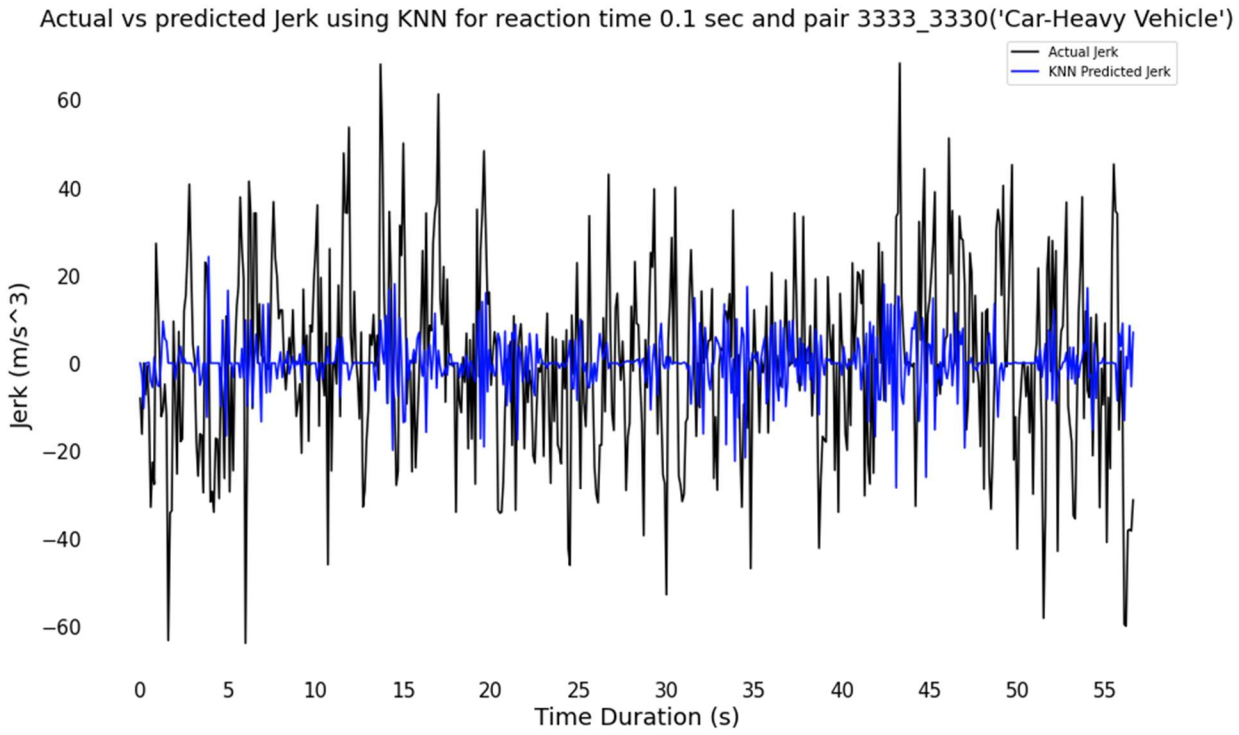

Figure 12. Jerk for 0.1 reaction time for KNN model

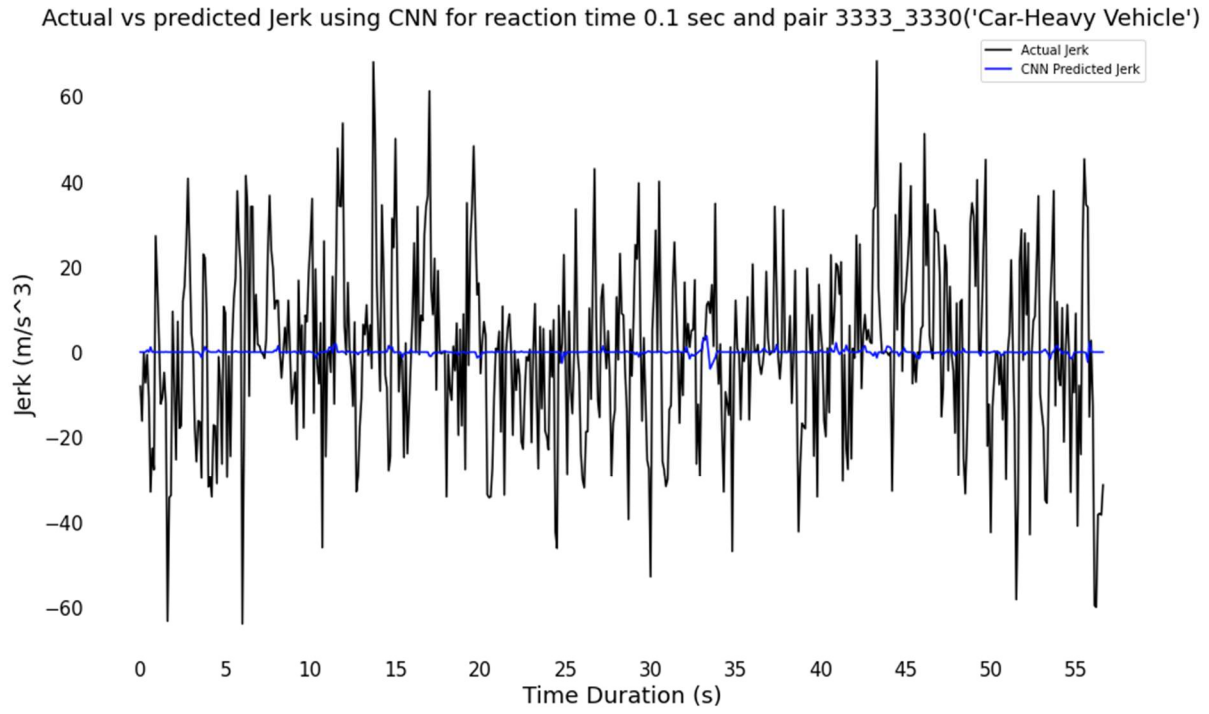

Figure 13. Jerk for 0.1 reaction time for CNN model

All the above tables (table 3, table 4, table 5, and table 6) show the R^2 and RMSE for all the three models for Acceleration, velocity, spacing, and jerk for the two highways I-80 and US-101 in the U.S.A. for the reaction time of 0.1 seconds. And the above figures (Figure 4 to Figure 11) give the trajectory plots comparing actual and predicted values for each model.

From the overall comparison of all the three models for different reaction times, it can be said that the R^2 and RMSE for CNN are better as compared to Random Forest and KNN. But when it comes to the trajectory plots, the Random Forest and KNN models have better trajectories. Overall, the models give good trajectory plots for Velocity and Spacing, but for Acceleration and Jerk, more work can be done to improve the trajectories.

Code Github Repo:
https://github.com/VineetDhamija/DataDrivenCarFollowing/tree/master

Conclusion

Assumption based models have a lot of limitations and can not be made generic as the driving situations changes from place to place. Because of this limitation researchers around the world are trying to create Data-driven car following models. In this project, Data Driven Car following Random Forest, KNN, and CNN models were trained with NGSIM dataset to predict acceleration and thus rest of the vehicle trajectory following a Lead vehicle in a single Lane. The result show that even though the R^2 and RMSE values of CNN is best among them all and should be used to predict the acceleration but the trajectories of the predictions with the actual values of the pairs say otherwise. This leads us to conclude

that the using R^2 and RMSE errors is an incorrect basis of model selection and usage. We should be using Trajectories as they are better depiction of the Human car driving based on real data. Both KNN and RF have shown better results that CNN, even though their Error are worse. Overall KNN performed best when validated for trajectories of the entire test set.

Future Work

We anticipate to take on the below challenges as an enhancement to the current work.

- Create Prod App which predicts based on excel input.
- Use Server and Docker to move past storage and CPU limitation.
- Once on Docker update Random Forest from 30 to 150 Regressors.
- Re-Train models on entire NGSIM data including Lankershim and Peachtree.
- Use Dynamic Time Warping to verify Accuracy.

Acknowledgement

We would like to thank U.S. Department of Transportation for collecting the NGSIM data and making it public. We hope that our project contributes toward the work going on in microscopic data-driven car following models.

This project would not have been made possible without the guidance and support of Prof. Umair Durrani at St. Clair College.

References

The *Data-Driven Car Following Model* has been made possible by referring to the below:

[1] "The number of cars worldwide is set to double by 2040 | World Economic Forum." https://www.weforum.org/agenda/2016/04/the-number-of-cars-worldwide-is-set-to-double-by-2040 (accessed Jul. 27, 2022).

[2] "The Most Common Causes of Car Accidents in Canada - Mann Lawyers." https://www.mannlawyers.com/resources/the-most-common-causes-of-car-accidents-in-canada/ (accessed Jul. 27, 2022).

[3] "Three Equations of Motion." https://simply.science/popups/2200.html (accessed Jul. 28, 2022).

[4] "Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data | Department of Transportation - Data Portal." https://data.transportation.gov/Automobiles/Next-Generation-Simulation-NGSIM-Vehicle-Trajector/8ect-6jqj (accessed Jul. 27, 2022).

[5] H. Shi, T. Wang, F. Zhong, H. Wang, J. Han, and X. Wang, "A Data-Driven Car-Following Model Based on the Random Forest," *World J. Eng. Technol.*, vol. 09, no. 03, pp. 503–515, 2021, doi: 10.4236/wjet.2021.93033.