Deep Learning based trajectory prediction for autonomous vehicles using V2V communication

1st Hardik Vaniya
MEng ECE. University of Waterloo
Waterloo, Ontario
hrvaniya@uwaterloo.ca, 20983430

2nd Vijiithaa Sasidharan
MEng ECE. University of Waterloo
Waterloo, Ontario
vsasidha@uwaterloo.ca, 20969795

3rd Shanthi Meena Arumugam
MEng ECE. University of Waterloo
Waterloo, Ontario
smarumug@uwaterloo.ca, 20986268

***Abstract*—As advancements in autonomous vehicles are becoming ubiquitous, a safer, accurate and well-predicted system has become most-desirable today. Because autonomous vehicles (AVs) must interact with other vehicles, it is critical to have a thorough understanding of the dynamic traffic environment, including future prospective vehicle trajectories. With the advent of autonomous vehicles, this ability has become even more important. Recent development in Vehicle to Vehicle(V2V) communication made it a crucial part of autonomous software solutions. In this project we are focusing on predicting leading-vehicle trajectory by applying deep-learning solutions on V2V communicated data from the coordinated vehicle along with driving style predictions. The proposed method incorporates vehicle data including speed, acceleration, time, jerk, steering angle in order to define driving style by an unsupervised clustering based algorithm. We use the LSTM-RNN network for trajectory prediction since it is best suited for time-series prediction and provides us with the desired output on time-stamped data from leading-following vehicle networks using V2V technology. Further we will combine output of driving style and LSTM prediction in deep neural network regression model for finalizing the leading vehicle path. In the provided architecture we are using a data set from US101 and I-80 highway data. Overall, we will use surrounding vehicle information to improve leading vehicle path trajectory recognition in order to improve vehicle autonomy.**

***Index Terms*—Trajectory prediction, V2V communication, LSTM-RNN**

I. Introduction

Autonomous vehicles have been seen as a viable solution to the issues of traffic congestion, road safety, and energy crisis since the 1980s. In the actual urban traffic scenario, autonomous vehicles still have a lot of driving challenges to overcome. Firstly, how to engage with various types of cars in a driving environment safely and sensibly is a significant challenge. And this is influenced by a number of circumstances, including the irrationality of driver, biker, and pedestrian behavior, intense interactions with other road users, and spatial restraints imposed by road geometry, the mobility of road users surrounding the AV(Autonomous Vehicle) is frequently challenging to predict. Human drivers with experience can forecast the future trajectory of other vehicles in a driving scene and make decisions that are safe, reasonable, and effective. At this time, a decision-making control has substituted an underlying control in the definition of an autonomous vehicle. An ideal autonomous driving system would be able to adjust the vehicle's mobility in real-time, responding to the surroundings and movement status as a driver would. Accurate trajectory prediction minimizes or eliminates the danger of an accident when AV performs challenging driving maneuvers including merging, changing lanes, and overtaking. It also improves the driver's comfort and productivity [2].

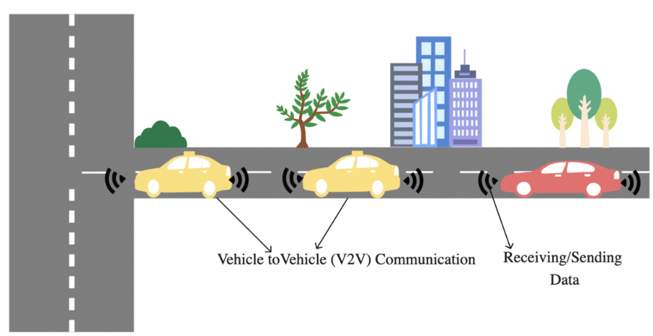

Fig. 1. V2V communication

The study and development of autonomous driving have made great strides thanks to recent developments in sensing and machine learning. The two basic methods for obtaining autonomous driving are as follows. The first is the end-to-end strategy, which uses a single model to directly map raw sensor data to control commands(speed, steering angle, brake etc.). The second method is the conventional engineering strategy [6], [7], which includes several modules like detection, tracking, prediction, and planning. Both strategies have benefits and drawbacks when it comes to AV safety. However, the traditional engineering approach is expected to dominate in the near future due to technological advances in the field and positive research outcomes. Predicting the trajectories of the AV's surrounding vehicles is a critical task of the traditional engineering method, as this knowledge is needed to ensure safe and reliable driving. Considering the next-generation road mobility, transportation is being built with connected cars, vehicle-to-vehicle(V2V) communication mechanisms come into the picture, with the goal of making transportation safer and more efficient[15]–[17]. The design of future intelligent and automated driving vehicles will also be strongly influenced by the V2V method.

Traditional methods for leading vehicle trajectory are often predicted using data collected from vehicle perception which is computationally costly to process. While data obtained from V2V communication will exchange additional information to help improve accuracy. Moreover, the advances in 5G technology are making vehicle to vehicle communication more efficient and robust.

The goal of this project is to develop an efficient and personalized motion prediction algorithm for connected vehicles based on the driving styles and predicting the trajectory of moving vehicles on highways, where cars and trucks make up the majority of the traffic. The human learning process is characterized by the continuous completion of related tasks using prior knowledge and experiences. The same learning scheme for connected vehicles towards high precision motion prediction is also developed in this study by utilizing prior motion states of the following and leading vehicle pairs.Various driving styles are initially defined by unsupervised learning; however, they can then be used to group vehicles with similar behaviors together. Then we have used the LSTM-RNN network to process the sequence data(motion state at every time stamp) which are collected via V2V communication. The obtained driving style and predicted vehicle state will then be fed into the regression neural network model for finalizing our future vehicle trajectory. We have followed the NGSIM(Next Generation Simulation) US -101 highway vehicle trajectory data-set which contains 45 minutes of data segmented into three 15 minutes periods.

The methods and methodologies utilized for predicting future motion will be the main topic of the literature review section that follows.

II. Literature Review

In [19] they propose Highway Trajectory prediction of surrounding vehicles with an LSTM network. It presents a consistent trajectory prediction by a Long short-term memory (LSTM) neural network, which predicts future longitudinal and lateral trajectories of vehicles on the highway. Mostly, the machine learning techniques can be classified between classification or regression methods. Here the approach is to train a predictor for the future trajectory of a single "target" vehicle. The fundamental benefit of behavior prediction is that discrete outputs make it simpler to train models and assess their efficacy, these predictions may then be utilized to plan the mobility of an autonomous vehicle.So [18] uses an artificial neural network in the form of a Long Short-Term Memory (LSTM) network as they are well suited for time series.They used the Keras framework, which implements the extended LSTM to achieve efficient results for the model.

For the target vehicle, they have defined features like i) local lateral position xtarg, to account for different behaviors depending on the driving lane. ii) local longitudinal position ytarg, to account for different behaviors when approaching the merging lane. iii) lateral and longitudinal velocities vxtarg and vytarg.The training and evaluating the model using the whole NGSIM US101 data-set without a-priority selection, the user can estimate future trajectories with an acceptable average RMS error 0.7 m (laterally) and 2.5 m s1 (longitudinally) when predicting 10s ahead. They wanted to create a network that can comprehend predictions for medium-term (up to 10 s) relationships. They are using windows of 100 inputs, which represent a total of 10s prior observations, to prevent back propagation-related problems that often occur with extended time series.

In [20] paper, a combination of maneuver-based and learning-based prediction models has been used to propose an enhanced trajectory prediction model on the basis of LSTM network driven by knowledge. At first, to describe the prior knowledge about a driving scenario, a driving knowledge base is first developed. Then, a rule-based online reasoning system is used to construct the prediction reference baseline (PRB) based on the driving knowledge base. Lastly, an LSTM neural network is used to forecast the target vehicle's future trajectory. The projected trajectory takes into account both posterior and past information without increasing computational complexity.

The proposed model categorizes and assesses the key factors that will influence a vehicle's trajectory in the future from the perspectives of safety, legitimacy, and reasonableness, a method of deterministic scene evaluation is used, which makes modeling of the spatial interactions easier. In order to increase the suggested model's versatility and accuracy, the Frenet coordinates based on the PRB have been utilized to train the LSTM network. The training data set does not need to be manually annotated in accordance with the particular driving situation. They have also used the (NGSIM) data-set in I-80 and US101 for model training and testing. The findings demonstrate that the suggested approach beats the state-of-the-art model and reduces the system's overall RMSE value by an average of 10Because of the unique combination of both the prior and posterior knowledge, the proposed model can adapt to different driving scenarios. The target vehicle's lane-changing behavior can be predicted on average 2.05 s (for LCL) or 2.71 s (for LCR). The precision can also be improved by 12.5 percent for long-term predictions and is more robust, flexible, and adaptive in complex traffic scenarios.

In [18]] Zhang, Z. and Ohya, J. have proposed a novel vehicle to vehicle(v2v) communication for improvised autonomous driving. Here they are taking advantage of the cooperative vehicles which are surrounded by the target vehicle. Using v2v target vehicles get data of motion state and data of driver's first view images from the surrounding vehicle which helps in improving the overall system performance.This also helps in gathering more temporal information, which combined with spatial data promote more robust outcome. To get best results they have introduced CNN + LSTM based architecture where CNN processes image data captured from the cooperative vehicle and LSTM will be calculating past motion data points to predict the future movement of the vehicle, combination of both the implementations perform better in most of the scenarios.They have describes movement control as below,

$$MC_V = \{steeringangel, speed\} \quad (1)$$

Where MC is movement control and V denoted as vehicle, as here in case of v2v communication, the surrounding environment is the state of our cooperative vehicle, the movement control(MC) at next step (t+Δ t) is defined as below.

$$MC_{auto}^{(t+\Delta t)} = \{MC_{auto}^{(t)}, MC_{coor}^{(t)}\} \quad (2)$$

Where "coor" is coordinated vehicle and movement control is described by two values steering angle and vehicle speed which are the final predicted output of the system.

Results of [18] suggest Res-Net to be the best model, transfer learning for CNN network performs well and the whole network is pre-trained on the ImageNet Data-set.On top of the CNN they have applied LSTM layers to capture more history data in the decision of future movement. Empirically, LSTM is a well improved RNN network which outperforms many prediction fields.[18] uses the Udacity data-set for the experiments. Which contains 223GB of image frames and 70 minutes of driving log data with the annotation of speed, longitude,latitude,throttle, brake, steering angles etc.

To conduct the accuracy and performance of the proposed system they have calculated the MSE(Mean Squared Error) of the steering angel and the car speed. And they got best MSE at following settings: with Cooperative vehicle, integrated LSTM, Res-Net CNN, 8 sequential frames, and 10-40 meters distance between host vehicle and cooperative vehicle turn out to be best performance model.

III. Contribution and Methodology

Many studies have tried to solve the vehicle trajectory problem and most of them have achieved state of the art results on this particular dataset. Despite the fact that many of them employ standard algorithms and use the same approach for trajectory prediction. The drawback of this strategy is that it will exclude several potential avenues for enhancing the outcomes. While designing, all the potential possibilities have been taken into account that humans may observe and use while making a decision to create a system that can mimic human thought. Hence, an effort to incorporate driving style behavior of the leading vehicle has been made in order to enhance the accuracy of our future trajectory prediction.

As stated, to develop this system NGSIM US-101 highway dataset has been used and it contains all the needed features. Training both of the models using selective features of the whole dataset (Data Preprocessing section goes into further detail on this). The whole architecture of the defined method can be represented

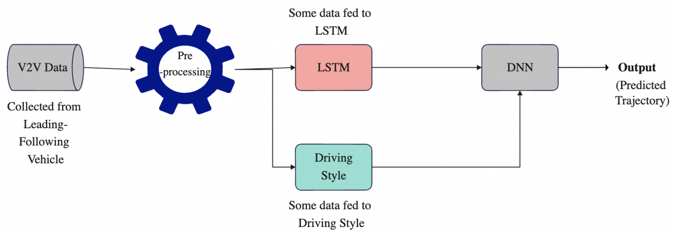

Fig. 2. General flow of our proposed method

as shown in Fig(2). For driving style prediction, unsupervised clustering algorithms have been applied to classify given driving information into one of the three types of driving styles such as Aggressive, Moderate and Normal. Dataset contains the trajectory of almost 1993 vehicles and each has its own driving trajectory which suggests driving behavior of the driver. Additionally, because the provided data are time-stamped, time-series clustering has been applied on the various characteristics of a particular trajectory or driving technique.

To ensure that our algorithm takes into account the time-shifted trajectory during training, DTW (Dynamic Time Warping)[21] technique has been used which essentially assesses how similar the two time series sequences can be time shifted. The training series data has been applied to TimeSeriesK-means, an unsupervised clustering technique, for classification, which basically takes all the input data and takes into account the number of clusters to classify. In this case, three clusters for each driving style prediction.

To predict a leading vehicle trajectory, LSTM model [22] has been used which is a state of the art model and a type of RNN for processing longer sequences. To train the model, following features such as local position of the vehicle and their global coordinates at each time stamp has been utilized. If trained on adequate historical data with future label output, LSTM performs better at learning patterns .Our network has to go through about 100000 steps of training to become strong.(Additional detailed explanation is given in section of LSTM). Finally, the driving style prediction will be used to improve the output for vehicle trajectory prediction for LSTM. This highly relies on the V2V communicated data from the surrounding vehicle. Therefore, both leading and following vehicle data has been used to make better predictions about the vehicle which is heading ahead and to know its future path to make autonomous judgments may be made with accuracy.

IV. Data processing

In this section, an explanation of the data processing has been done in order to train the driving style model and LSTM model. Here, Data processing contains both data pre-processing part to smooth out the data and post processing part to make it suitable for training defined models. First, we will get a detailed understanding of the used dataset.

The description of the steps can be seen in Fig(3)

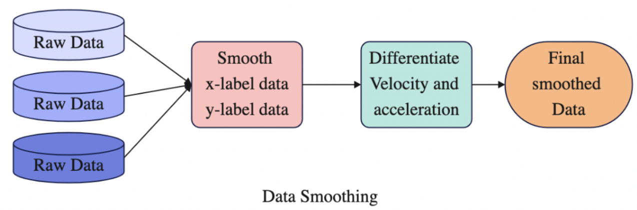

Fig. 3. Data Smoothing

In this project , NGSIM US-101 trajectory dataset has been used and the same has been released in 2005. This dataset has been extracted from video recordings of 8 cameras mounted over buildings looking towards Hollywood Freeway in California.

The video' frames and other vehicle details have been extracted using NG-VIDEO software.This is a well labeled dataset with every frame id and vehicle ids and the same has been useful to identify related velocity and acceleration of the vehicles. Minimum frames for each vehicle are 158 which is enough for getting the initial trajectory.However, several researchers discovered that this dataset contains numerous anomalies, including abrupt changes in velocity, acceleration, and deceleration..[23]. To overcome this drawback, smoothing of the dataset can help in the processing and training process and a very good implementation of smoothing on this particular data has already been provided by Rim-EL-Ballouli [24]. He has used Savitzky-Golay Filter to make data smooth and less-noisy. As per his approach, the smoothing is done in two steps. First the author applied smoothing on X-local and Y-local(according to the California state plane coordinate system) coordinates and then he has computed the values of velocities and acceleration with respect to the newly obtained smoothed X-local and Y-local values.

To avoid the smoothness process, we have utilized already given smoothed data. The dataset has 20 feature columns and all of them are not needed to train the model. So eight features have been selected out of all

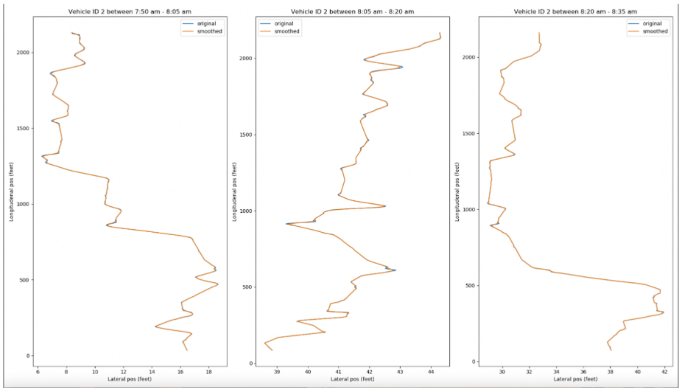

Fig. 4. Smoothed Data

the features to further process our data and have been fed to the model training. The description of the features is provided in the Fig(5)

Vehicle Id	Vehicle Identification Number
Global Time	Global time in milliseconds
Local X	X Coordinate of the vehicle in the frame
Local Y	Y Coordinate of the vehicle in the frame
Vehicle Velocity	Current velocity of vehicle in feet/second
Vehicle Acceleration	Current acceleration of vehicle in feet/second square
Space Headway	Front distance of the vehicle from preceding vehicle
Time Headway	Time in seconds needed to cover the front distance

Fig. 5. Features of NGSIM US-101 Dataset

After thorough analysis, a few of the selected features have been utilized to train the models for both Future Trajectory Prediction and Driving Style Recognition. Detailed explanation of the same, can be found in the next sections.

V. Driving style Recognition

Clustering algorithms are the best when patterns have to be recognised out of the input data and assign similar patterns in the same cluster. With unsupervised training, our model can be trained without input labels. It will learn to recognise similar patterns at every input step and organize itself to classify new input into one of the obtained clusters at the end of the training.

In our project, a well known k-means clustering algorithm[28] has been used. K-means essentially aims to partition given inputs (which could be of n dimensions) into K clusters by minimizing within cluster distance and maximizing inter-cluster distances. The mathematical representation of the above idea[27] has been interpreted below.

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i \tag{3}$$

where μ_i is the mean of points in Si and Si is the defined cluster set.

Driving style can be classified into three classes i) Aggressive ii) Moderate and iii) Normal. which may be determined from the features collected from the vehicle and also its behavior from each time stamp. While drawing inference, v2v communication will be used to collect those facts.. There is a need for specific data from the list given in Fig(5) like vehicle velocity, acceleration, time and space headway. All these mentioned features are enough to decide driving behavior of any driver on the highway. For example, an aggressive driver will have frequent sudden breaks and acceleration and mostly have less time headway from the front vehicle.

Depending on the above analysis, training has been performed over the k-means model by following two steps.

1) Pre-processing the data to make it suitable to k-means
2) K-means training for three clusters(for Aggressive, Moderate and Normal Driving style)

1) Pre-processing of the data : It is important to gather enough driving styles in the form of numeric data to classify them into similar clusters. In order to achieve that, first collect all the given data and then using their vehicle id aggregation in a different cell. After separating each feature according to its vehicle id (about 1993 vehicles), the mean of all the features has been considered and stored the same in the new database. The newly acquired features are shown in Fig(6).

Features	Statistical Operation
Vehicle Velocity	Mean - Average Velocity
Vehicle Acceleration	Mean - Average Acceleration
Space Headway	Mean - Average Space Headway
Time Headway	Mean- Average Time Headway

Fig. 6. Statistical Operation

This new dataset contains average(mean) values of selected features for 1993 different vehicles which can then be used to train our clustering algorithm. As there is only a maximum of 1000 frames per car, 100 seconds of the trajectory of the vehicle is more than adequate to identify the driving style. And if the average of the obtained features has been taken into account, algorithm will provide the collective parameters for the driving style and the same has been further fed to the k-means which then will further divide into three clusters.

2) Training k-means Algorithm: Figure(7) shows the visual steps followed while the training of k-means for two clusters. (a) has all the training data that needs to be divided in two clusters. As an initial step, a number of clusters have to be assigned, in this case it is two which eventually become the number of centroid points in the input space as shown in (b).

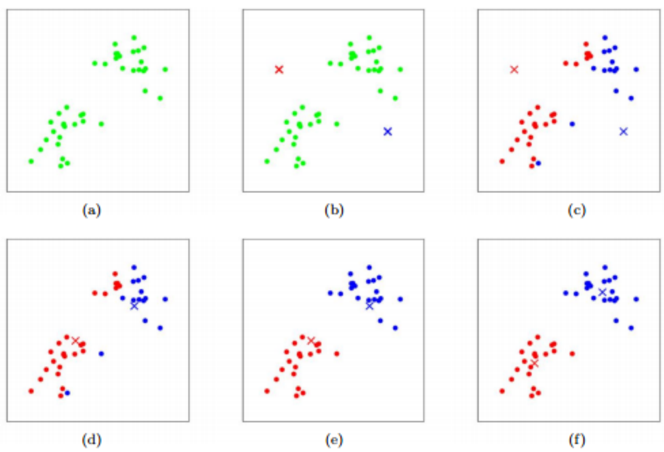

Fig. 7. K-means training Steps [29]

While training at each input step these centroids keep updating using the defined distance measure. Upcoming members will be assigned one of the defined classes(centroid) and again the new centroid will be calculated depending on the clustered points. (c) and (d) depicts the explained process. At the end of the training process or when the last data point encountered, k-means will have two different centers which can be easily classified with a hyperplane in a given space.

After the training, during inference the model calculates the distance between the cluster head and the unseen data point in the same dimension space and assigns a class which is closest among all centers.

In our project, we have used k-means to cluster our 1993 vehicle ids into three classes using features given in Fig(6). To train the model, the value of K has been provided which is essentially the value of the number of clusters needed to train the data for. The accuracy of the model will highly depend on the value of K chosen. In this project, as the end goal is to train the model to recognize these three driving styles, K = 3 value has been chosen to train the model. While training at each iteration(new data point) k-means, find the best centroids by assigning the data points to a particular cluster and calculating distances to update the cluster ranges. This process will keep going until the last training data point or the model converges and there is no change in cluster centroids.

To train k-means on different features values and to improve the results, other two features have been included like local-X and Local-Y co-ordinates which are relative positions of the vehicle at each time stamp. Results showed that, as the mean of the features has been taken into consideration, the mean of X and Y coordinates leave their effect in the driving style prediction while the earlier four features were very well able to classify data into three clusters.

VI. LSTM-DNN Architecture

Second and foremost important part of this project is to train a sequence prediction model to predict future trajectory after fetching past input data points. In this section, the implementation of LSTM and its working has been explained.

LSTM(Long short term memory) is a type of RNN model which came out as a real breakthrough in the sequence modeling technology. The use case of RNN model is to learn dependency of the input sequences on each other and also a relation between the words in a sentence. The problem with conventional RNN is that it fails to remember long sequences and its dependencies. To overcome this, LSTM model was introduced, which is having memory blocks and a forget gate to keep the context of long sentences and to forget the redundant words. The figure of the single LSTM block is given in Fig(8)[31].

LSTM has a total of three gates to propagate the information to the next state such as 1) Input gate 2) Forget gate 3) Output gate. Architecture shown in Fig(7) constructs a chain-like deep model which takes input from input gate, processes the input and stores the important value in cell state C while forgetting the unnecessary values via "forget gate" operation. Output of current hidden state and cell C feed to the next block's hidden state, that is how it keeps the importance of past inputs.The operation of LSTM blocks are given below.

The value $\tilde{C}_t$ is the candidate cell state which can be represented as:

The cell state ct of the LSTM unit is the combination of previous C_{t-1} and current candidate states.

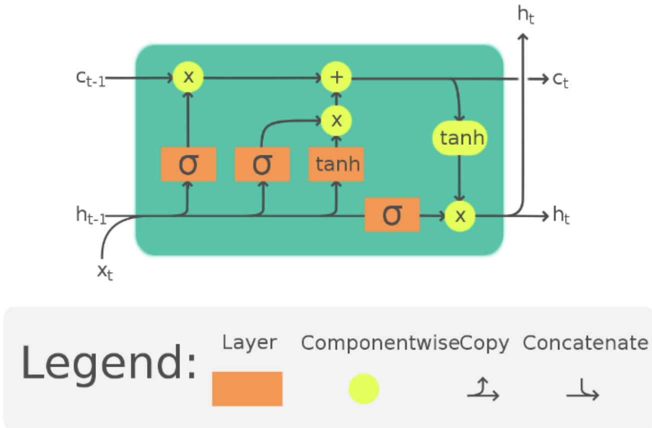

Fig. 8. Single LSTM Block

$$
\begin{aligned}
f_t &= \sigma(\mathbf{U}_f x_t + \mathbf{W}_f h_{t-1} + b_f)(1) \\
i_t &= \sigma(\mathbf{U}_i x_t + \mathbf{W}_i h_{t-1} + b_i)(2) \\
o_t &= \sigma(\mathbf{U}_o x_t + \mathbf{W}_o h_{t-1} + b_o(3)
\end{aligned}
$$

$$\tilde{C}_t = tanh(\mathbf{U}_o x_t + \mathbf{W}_c h_{t-1} + b_o)$$

$$C_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

Finally, the cell outputs the products of the cell state C_t and the candidate output from the output gate.

where σ is the activation function, and in most of the times it is sigmoid function only. x_t and h_{t-1}, are the current input vector and the previous hidden state. f , i, o are the forget gate, input gate, and output gate. U, W, b are corresponding model parameters.

A. Proposed Approach for trajectory prediction using LSTM

For this project, **LSTM + DNN** (LSTM- Deep Neural Network)layers has been used. For the dataset, pre-processed data has been obtained from the processing layer. The used features from the dataset are given in the Fig(9).

Here unlike Driving Style Recognition, all the sequential data points are required for all the vehicles at each frame to get the whole perspective of the past pattern. Also, Local coordinates of the vehicle plays major role in the prediction of the future trek and therefore the following Local_X and Local_Y features has been included from the NGSIM dataset. The processed dataset has 1000000 rows and around 8 important features.

A lot of processing power is needed to train this enormous data, there was limited resources to train our model on 100,000 rows of data. Because of which, six features out of eight have been necessary for the trajectory determination.To make all features on the same patch feature scaling have been performed on the whole training data set. Scaling makes data centralize to the value zero, and scale all the values in a particular range. For any deep learning algorithm scaling is an important part as it helps generalize while training the model. This data transformation approach is easy to implement and it is convenient to do inverse transform after prediction.

Preparing the dataset compatible with the LSTM input is also a crucial task as the performance of the sequence prediction model highly depends on the number of past inputs taken into account. In order to get the best number several combinations of past input sizes have been tested and 8 past inputs were providing best results. Therefore, batches of 8 past points have been created as input to the model and current 9th data point as the label. After the data is prepared as a suitable input format, the shape of input data = (99992, 12, 1) and the shape of output labels = (99992, 1, 2). The architecture for our LSTM-DNN is presented in Fig(10).

As shown in the architecture, one LSTM layer has been used with 32 nodes and the input size is the shape of the single input data. "relu" activation function has been applied at the output of the LSTM layer to keep output values bounded. After that a dropout layer has been added with a drop out rate of 0.5 to keep the model from over fitting. Next layer in the architecture is a dense layer with 16 neurons and a relu activation function. Another dropout layer has been added with the rate of 0.2 for the same purpose. The last layer of the model architecture is to generate two values of X and Y coordinates for the vehicle. This will be predicted values from the given input features maps

The model has been trained with two different optimisers, known as "Adam" and "RMSprop" with the loss function as MSE(mean squared error). Therefore, the total learnable parameters after this setting is around 5042. To evaluate this model, the evaluation set of the training data has been kept at 0.1%. As this is a

$$h_t = o_t * tanh(C_t)$$

Features	Local_X	Local_Y	Vehicle Velocity	Vehicle Acceleration	Space Headway	Time Headway

Fig. 9. Features

sequential learning, the batch size is 1 which signifies a single data point has to be fed for every iteration. The whole procedure lasts for 5 epochs.

VII. Results

In this section, all the benchmark and evaluation methods to explain the results have been presented. The metrics used for the evaluation of the results are given below.

MSE(Mean Squared Error) : This is a matrix which will give the total error between the actual values of co-ordinates and the predicted values. It can be represented by the mathematical formula below.

LSTM Architecture : With different values of nodes, the LSTM layer has been added and the model has been trained.

Epoch : To improve the model performance, model has been trained on various number of epochs.

A. Result discussion of driving style

Once the training has been done, the Unsupervised driving style prediction mode has been used to store the clustered data into a table and the mean has been calculated from the newly classified data to get a better idea of our classification results. The mean values of the classified vehicle features as aggressive, moderate and normal has been presented in the Fig(11).

From the table it can be figured out that for Aggressive style the value for Average Time Headway is 8.06 while Moderate and Normal driving styles have values around 534.06 and 1132.90 respectively. Given that aggressive drivers frequently cross the distance more quickly, this seems logical. Same way values of average velocity for aggressive style is more than(almost 1.5 times) the moderate style.Though the difference between normal and moderate style velocities is very less which can be due to some similarities between moderate and normal driving style.

To see the clustered output, dimension reduction methodology has been used in this paper. As we have

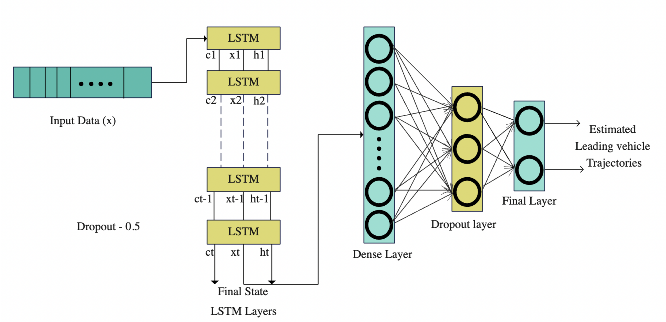

Fig. 10. LSTM-DNN Architecture

$$MSE = \frac{1}{n}\sum_i^n (y_i - \hat{y}_i)^2$$

used four features for training k-means and it is not possible to plot the 4D figure, **PCA(principal component analysis)** algorithms have been used to get two most prominent features out of four used features. Fig(12) show the representation of the clustered classes of driving styles. Three clusters can be clearly identified from the 2d presentation of the trained k-means output.

B. Results from LSTM-DNN mode

In this paper, training of **LSTM-DNN**architecture has been performed on the metrics which are defined above. Different combinations of the hyper parameters have been applied while training the model and the best obtained result is 0.0207 MSE. Fig(13) depicts the MSE error for various settings of parameters on 100000 rows of data with "Relu" activation function and "Adam" optimizer. The best MSE setting we got for 7 epochs and 8 past time steps values. To measure the performance of our model two parameters, training loss and validation loss have been observed.

As shown in Fig(14) it is decreasing with the number of epochs. But after a few epochs validation loss seems to increase. In order to get the best model out of each trained epoch, checkpoints structures have been introduced. It will store the best trained model based on evaluation parameter which in this case is validation loss.

In Fig(15)we have presented a predicted future trajectory on the given input data and the original trajectory to compare the model performance, and as it can be seen that it performs well when the route of the vehicle is less fluctuating and it shows some differences when there are sudden turns.

Driving Style	Id Count	Avg Vel	Avg Acc	Avg Space H	Avg Time H
Aggressive	1818	41.880956	0.427515	81.162913	8.065465
Moderate	106	28.258050	0.123958	67.078214	534.086185
Normal	69	27.038102	0.177454	64.364374	1132.907198

Fig. 11. Classified average feature values for each driving. style

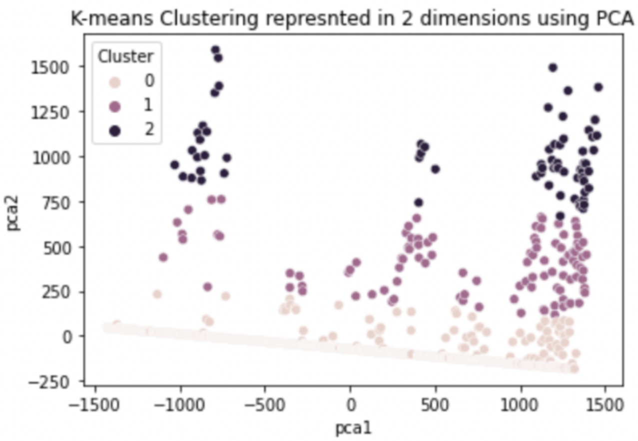

Fig. 12. Driving Style Cluster Representation

In the future, to further expand this project we can incorporate building of different Neural Network architecture according to the predicted driving style.This can add more perspective to the future trajectory prediction and can give better results. Also more and diverse data from various scenarios can be gathered and used to train our proposed model to make it more generalized and robust to the surrounding environments. Also with the advent of the V2V technology more surrounding information other than vehicle data can help take the vehicle maneuver decision in real time.

VIII. Conclusion

In this project, a vehicle trajectory prediction system based on unsupervised driving style prediction and LSTM-DNN approach has been proposed. By analyzing data gathered which can be easily communicated through V2V technology and incorporating the driving behavior the future path prediction of the vehicle can be improved. Driving style prediction system designed in such a way that it can generate one of three styles. This prediction will be utilized in the LSTM predicted output to further make the correction in the model performance. The results have been analyzed based on the MSE(mean squared error) matrix and it achieved around 0.0278

Past time steps	MSE Val Loss	MSE train Loss	Epochs	LSTM Nodes	Activation
12	0.0329	0.1169	3	64	Relu
8	0.0262	0.1162	4	32	Relu
8	0.0235	0.0	2	32	Relu
8	**0.0207**	**0.0878**	**7**	**64**	**Relu**
8(RMSE)	0.1989	0.3060	3	32	Relu

Fig. 13. Comparison of MSE for different combinations of parameters

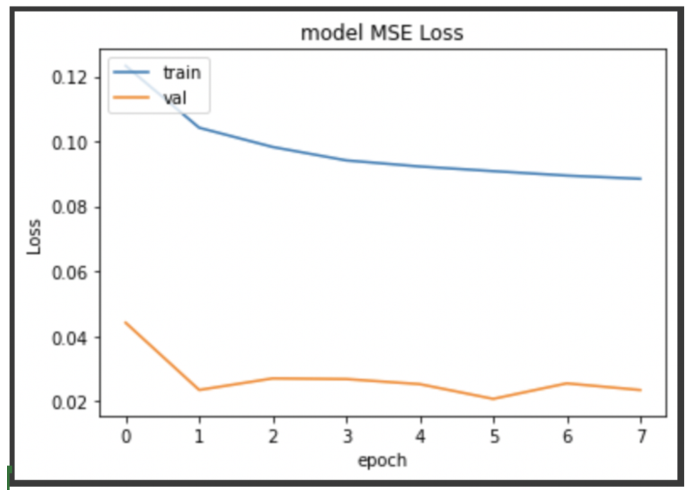

Fig. 14. Performance of the model based on MSE loss

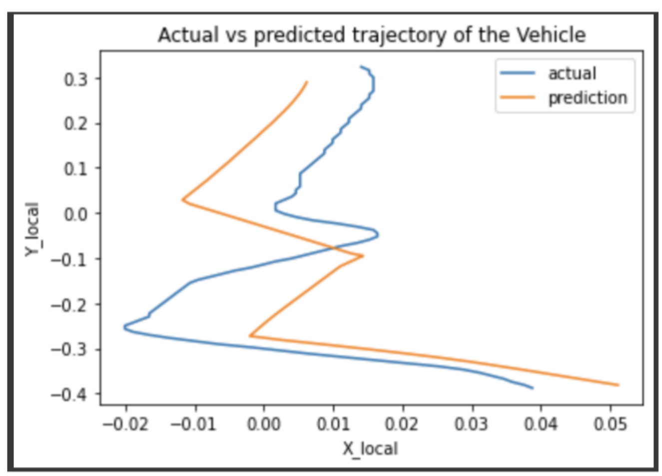

Fig. 15. Representation of Actual Vs Predicted Trajectory Prediction

feet of MSE error. Which is outperforming in various scenarios. This project's performance depends on the quality of information collected from the surrounding vehicle via V2V communication and in order to maintain that in future more information from various vehicles can be collected and processed to improve the performance of the model.

REFERENCES

[1] D. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," in Advances in neural information processing systems, 1989, pp. 305–320.

[2] G. Seetharaman, A. Lakhotia, and E. P. Blasch, "Unmanned vehicles come of age: The DARPA grand challenge," Computer, vol. 39, no. 12, pp. 26–29, 2006.

[3] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang et al., "End to end learning for self-driving cars," arXiv preprint arXiv:1604.07316, 2016.

[4] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp, "Off-road obstacle avoidance through end-to-end learning," in Advances in neural information processing systems, 2005, pp. 739–746.

[5] W. Li, D. Wolinski, and M. C. Lin, "ADAPS: Autonomous driving via principled simulations," in IEEE International Conference on Robotics and Automation (ICRA), 2019, pp. 7625–7631.

[6] W. Li, D. Wolinski, and M. C. Lin, "ADAPS: Autonomous driving via principled simulations," in IEEE International Conference on Robotics and Automation (ICRA), 2019

[7] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," The International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, 2013. [7] Y. Shen, W. Li, and M. C. Lin, "Autonomous driving via multi-sensor perception and weighted inverse reinforcement learning," Technical Report, University of Maryland, College Park, 2020.

[8] M. Zhou, X. Qu, and X. Li, "A recurrent neural network based microscopic car following model to predict traffic oscillation," Transportation research part C: emerging technologies, vol. 84, pp. 245–264, 2017.

[9] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng, and F.-Y. Wang, "Capturing car-following behaviors by deep learning," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 3, pp. 910–920, 2017.

[10] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp.1468–1480. , 2018

[11] X. Huang, J. Sun, and J. Sun, "A car-following model considering asymmetric driving behavior based on long short-term memory neural networks," Transportation research part C: emerging technologies, 2018.

[12] F. Altche and A. de La Fortelle, "An LSTM network for highway ´ trajectory prediction," in 20th International Conference on Intelligent Transportation Systems, 2017, pp. 353–359.

[13] "Next generation simulation (NGSIM)," https://ops.fhwa.dot.gov/ trafficanalysistools/ngsim.htm.

[14] S. Lefevre, D. Vasquez, and C. Laugier, "A survey on motion prediction ‘ and risk assessment for intelligent vehicles," ROBOMECH journal, vol. 1, no. 1, 2014.

[15] J. Harding et al., "Vehicle-to-vehicle communications: Readiness of V2V technology for application," National Highway Traffic Safety Administra- tion, Washington, DC, USA, DOT HS 812 014, 2014.

[16] B. Zhao et al., "Driving-cycle-aware energy management of hybrid electric vehicles using a three-dimensional Markov chain model," Automot. Innov., vol. 2, pp. 146–156, 2019.

[17] Z. Wanzhong, W. Gang, W. Chunyan, Y. Leiyan, and L. Yufang, "En- ergy transfer and utilization efficiency of regenerative braking with hy- brid energy storage system," J. Power Sources, vol. 427, pp. 174–183, 2019.

[18] Zhang, Z. and Ohya, J. "Movement Control with Vehicle-to-Vehicle Communication by using End-to-End Deep Learning for Autonomous Driving".

[19] F. Altché and A. de La Fortelle, "An LSTM network for highway trajectory prediction," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017, pp. 353-359, doi: 10.1109/ITSC.2017.8317913.

[20] Wang, Shaobo; Zhao, Pan; Yu, Biao; Huang, Weixin; Liang, Huawei, "Vehicle Trajectory Prediction by Knowledge-Driven LSTM Network in Urban Environments"Journal of Advanced Transportation,Hindawi

[21] Senin, Pavel. (2009). Dynamic Time Warping Algorithm Review.

[22] Hochreiter, Sepp Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[23] Montanino, Marcello, and Vincenzo Punzo. "Making NGSIM data usable for studies on traffic flow theory: Multistep method for vehicle trajectory reconstruction." Transportation Research Record 2390.1 (2013): 99-111.

[24] https://github.com/Rim-El-Ballouli/NGSIM-US-101-trajectory-dataset-smoothing

[25] Thiemann, Christian, Martin Treiber, and Arne Kesting. "Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data." Transportation Research Record 2088.1 (2008): 90-101.

[26] Altché, Florent, and Arnaud de La Fortelle. "An LSTM network for highway trajectory prediction." 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017.

[27] "K-means clustering" https://en.wikipedia.org/wiki/K-means-clustering

[28] Lloyd, Stuart P. (1957). "Least square quantization in PCM". Bell Telephone Laboratories Paper. Published in journal much later: Lloyd, Stuart P. (1982). "Least squares quantization in PCM"

[29] CS221, (July 31, 2022) https://stanford.edu/ cpiech/cs221/handouts/kmeans.html

[30] Mishra, SidharthSarkar, Uttam Taraphder, Subhash Datta, Sanjoy Swain, Devi Saikhom, Reshma Panda, Sasmita Laishram, Menalsh. (2017). Principal Component Analysis. International Journal of Livestock Research. 1. 10.5455/ijlr.20170415115235.