Western Michigan University
ScholarWorks at WMU

Master's Theses Graduate College

4-2020

Vehicle Velocity Prediction Using Artificial Neural Networks and Effect of Real-World Signals on Prediction Window

Tushar Dnyaneshwar Gaikwad
Western Michigan University, tushar.d.gaikwad@gmail.com

Follow this and additional works at: https://scholarworks.wmich.edu/masters_theses

Part of the Electrical and Computer Engineering Commons, and the Navigation, Guidance, Control, and Dynamics Commons

Recommended Citation

Gaikwad, Tushar Dnyaneshwar, "Vehicle Velocity Prediction Using Artificial Neural Networks and Effect of Real-World Signals on Prediction Window" (2020). *Master's Theses*. 5137.
https://scholarworks.wmich.edu/masters_theses/5137

This Masters Thesis-Open Access is brought to you for free and open access by the Graduate College at ScholarWorks at WMU. It has been accepted for inclusion in Master's Theses by an authorized administrator of ScholarWorks at WMU. For more information, please contact wmu-scholarworks@wmich.edu.

VEHICLE VELOCITY PREDICTION USING ARTIFICIAL NEURAL NETWORKS AND EFFECT OF REAL-WORLD SIGNALS ON PREDICTION WINDOW

by

Tushar Dnyaneshwar Gaikwad

A thesis submitted to the Graduate College
in partial fulfillment of the requirement
for the degree of Master of Science in Engineering
Mechanical Engineering
Western Michigan University
April 2020

Thesis Committee:

Zachary Asher, Ph. D., Chair
Richard Meyer, Ph. D.
Alvis Fong, Ph.D.

Copyright by
Tushar Dnyaneshwar Gaikwad
2020

VEHICLE VELOCITY PREDICTION USING ARTIFICIAL NEURAL NETWORKS AND EFFECT OF REAL-WORLD SIGNALS ON PREDICTION WINDOW

Tushar Dnyaneshwar Gaikwad, M.S.E.

Western Michigan University, 2020

Prediction of vehicle velocity is essential since it can realize improvements in the fuel economy/energy efficiency, drivability, and safety. Many publications address velocity prediction problems, yet there is a need for the understanding effect of different signals for the prediction. There are numerous new sensor and signal technologies like vehicle-to-vehicle and vehicle-to-infrastructure communication that can be used to obtain comprehensive datasets. Several references considered deterministic and stochastic approaches that use the datasets as input to determine future operation predictions. These approaches include different traffic models and artificial neural networks such as Markov chain, nonlinear autoregressive model, Gaussian function, and recurrent neural network. In this research, we developed different neural networks and machine learning algorithms that use different groups of datasets collected in Fort Collins, Colorado. Synchronous data was gathered using a test vehicle equipped with sensors along the drive route. The custom dataset consists of ego vehicle position, current velocity, ADAS-derived near-neighbor relative position, infrastructure-level transit time, and Signal Phase and Timing (SPaT). The effect of different groups of datasets on future velocity prediction windows of 10, 15, 20, and 30 seconds was studied. The results are assessed based on MAE and time shift. This research shows a 10-second prediction horizon that the lowest Mean Absolute Error (MAE) and the time shift of future velocity prediction. GPS, current vehicle velocity, and Signal Phase and Timing (SPaT) were the most influential parameters for prediction accuracy. Artificial neural networks performed better in terms of getting lower MAE while the classical machine learning model performed better in terms of the getting lower time shift.

ACKNOWLEDGMENTS

I want to thank Dr. Zachary Asher of the Mechanical and Aerospace Engineering Department at Western Michigan University for granting the opportunity to work on this project. This project would not have been possible without the constant support and valuable inputs from Dr. Zachary Asher. I want to thank Dr. Richard Meyer and Dr. Alvis Fong for supporting us as the thesis committee. We would also like to thank our collaborators from Colorado State University.

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE). The specific organization overseeing this report is the Vehicle Technologies Office under award number DE-EE0008468.

Tushar Dnyaneshwar Gaikwad

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Intelligent Transportation system (ITS)

The shift toward Intelligent Transportation Systems (ITS) will be the most disruptive since the initial days of automobiles[1]. It has the potential to completely transform the movement of people and goods, enabling safer and smarter transportation[2]. The ITS consists of several technologies such as Advanced Driver Assistance System (ADAS), Automated Driving Functions (ADF), Vehicle to Vehicle(V2V) and Vehicle to Infrastructure (V2I) communication [3].ITS also incorporates both wireless and wireline communications-based information and electronics technologies. Wireless technology is used to connect vehicle information and location to other vehicles, other transportation modes (such as pedestrians or bicyclists), local infrastructure, and remote infrastructure in the cloud.

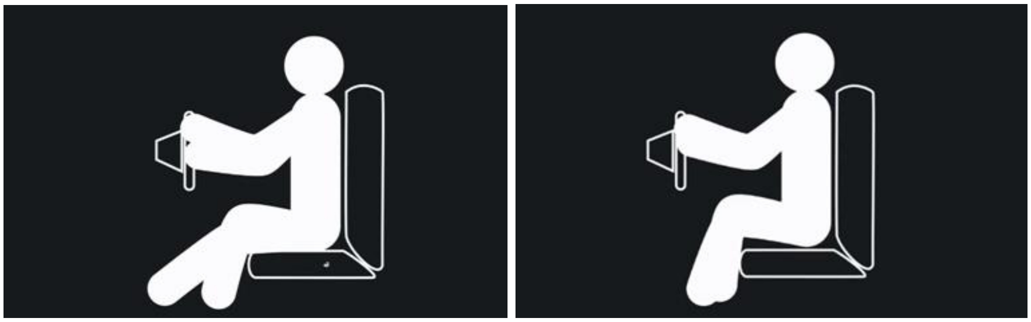
IDC forecasts that worldwide spending in 2017 on connected vehicles will be $29.6 billion, and government spending on intelligent transportation systems will be $16.5 billion [4]. However, obstacles to ITS adoption include sluggish growth in global infrastructure and high installation costs.

1.2. Autonomous Vehicle Technology

Autonomous cars could revolutionize the way we travel, giving us more time to work, play, learn, and relax on the road. Even if advancement in automation, most cars still have a human driver. SAE International has divided autonomy into five stages [5], which are shown [6] in Figure 1.

Figure 1: Stages of automation

Level 0 has No automation. The full-time performance by the human driver of all aspects of the dynamic driving task

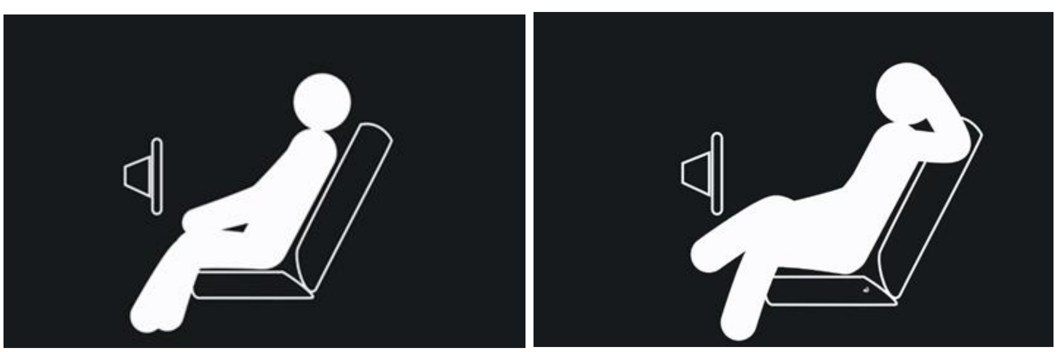


(a) (b)

Figure 2:(a) Driver assistance (b) partial automation

Level 1 is the driver assistance [6] is shown in Figure 2(a). The driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration

Level 2 is partial automation [6] shown in Figure 2(b). The driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration. Here the car can do some of the steering, braking, and accelerating, but still needs a driver with hands on the wheel, because level 1 and 2 are still just driver support. Both level 1 and level 2, use information about the driving environment and with the expectation that the human driver performs all remaining aspects of the dynamic driving task.



(a) (b)

Figure 3:(a) Conditional automation (b) high automation

Level 3 is conditional automation [6], is shown in Figure 3(a), which means that the car is in control, but requires human intervention in an emergency, or when prompted by the system. So, lane keeping, collision warning, even active interventions that will swerve the vehicle if it is about to get into an accident. At this level, the vehicle operates with the expectation that the human driver will respond appropriately to a request to intervene.

Level 4 is high automation [6], shown in Figure 3 (b). The driving mode-specific performed by an automated driving system of all aspects of the dynamic driving task. At this level, the vehicle operates even if a human driver does not respond appropriately to a request to intervene.

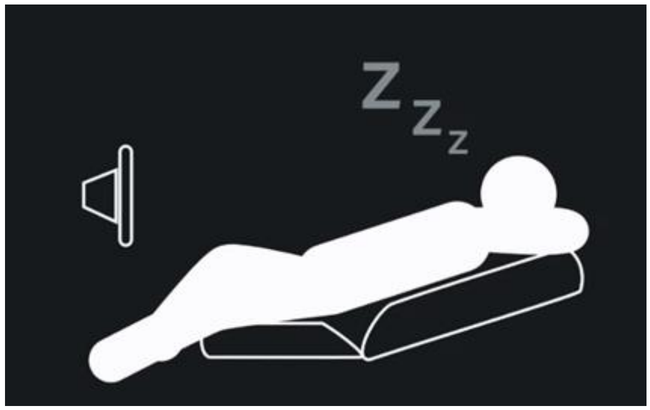


Figure 4: Full automation

The ultimate self-driving car would be operating at level 4 or 5, shown in Figure 4, where it can steer, brake, accelerate, monitor the road, respond to random events, choose to change lanes, and turn by giving blinker. At this level, the vehicle operates under all roadway and environmental conditions that can be managed by a human driver.

1.3. Motivation for the Project

1.3.1. Fuel Economy

In 2016, the transportation sector became the top contributor to greenhouse gas emissions, which eventually contribute to climate change [7]. Climate change is projected to significantly affect human health, the economy, and the environment in the world, particularly in futures with high greenhouse gas emissions and limited or no adaptation. Recent findings reinforce the need for substantial and sustained reductions in greenhouse gas emissions, and regional adaptation efforts. Without these methods, there will be significant and far-reaching changes over the 21st century with negative consequences for a vast majority of sectors, particularly towards the end of the century [8]. These accelerating emissions are putting the world on track to face some of the most severe consequences of global warming sooner than expected [9]. Figure 5 shows the effect of global warming around the world. To mitigate climate change, most recently, the Paris Agreement of 2015 took on the long-term aims. It requires holding the increase in the global average temperature to well below 2°C above pre-industrial levels and pursuing efforts to limit the temperature increase to 1.5°C above pre-industrial levels [10]. Because of these issues, governments around the world have imposed various Fuel economy (FE) requirements that automotive manufacturers are required to meet [11].

Figure 5: Effect of climate change

In Autonomous vehicle technology, one of the strategies proposed to minimize fuel consumption is Optimal Energy Management. Hybrid Electric Vehicles (HEVs) and Plug-in Hybrid Electric Vehicles (PHEVs) can increase the precision of vehicle operation and automation with the usage of the future prediction horizon. Optimal Energy Management Systems (Optimal EMS) adjusts powertrain operation proactively in connected and autonomous vehicles (CAV) [12]. With the usage of correct automation and accurate future operation prediction, HEVs and PHEVs can control the operations of a vehicle precisely.

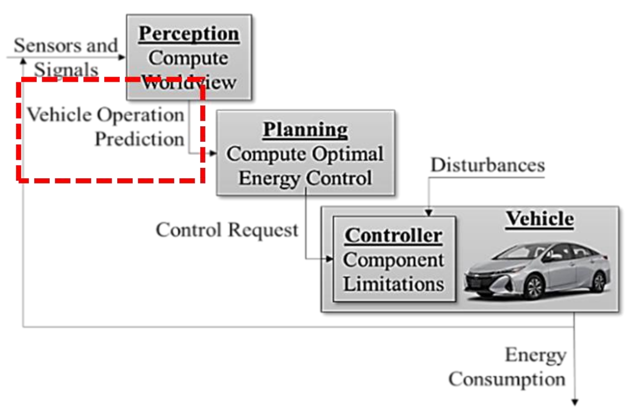


Figure 6: Optimal energy management strategy

Therefore, CAVs can adjust vehicle powertrain operation proactively using Optimal EMS. Figure 6 shows the Optimal Energy Management Strategy, where the dashed line highlights the input of future velocity prediction to the strategy. This example shows that the prediction of future driving conditions essentially determines the constraints of the energy optimization problem [14]. Due to such dependency between prediction horizon and Optimal EMS, there is a pressing need to develop accurate and robust approaches for predicting vehicle speed. It is also essential to understand which signals would produce better results.

1.3.2. Safety

In 2017, over 37,000 people killed and injured in US fatal collisions. These accidents often originate from driver inattention or impairment shown in Figure 7. More than 90 % of the time, the driver error is responsible for crashes in the U.S. Hence, autonomous vehicles must follow a robust design and validation process.

Figure 7: Crashes due to driver inattention

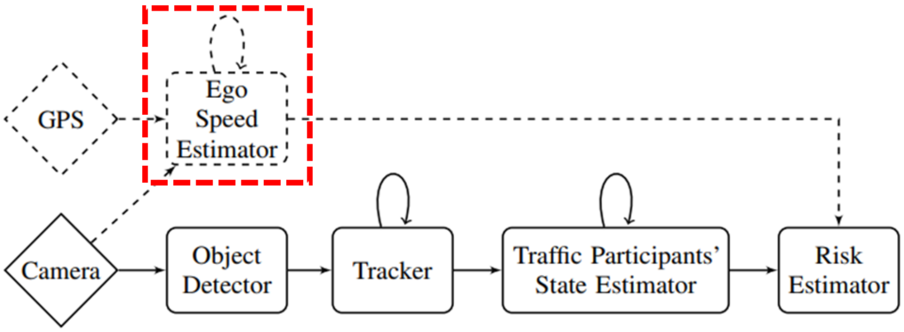
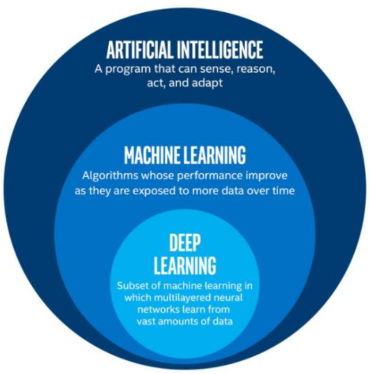


Figure 8: Collision risk estimator strategy

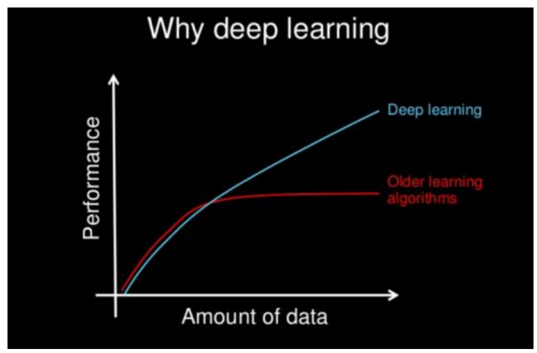
The Strategy proposed for increasing the safety of autonomous vehicles, as shown in *Figure 8*, where the dashed line highlights the input as velocity prediction to the strategy. The system is designed to be modular with interchangeable components for each of the main tasks. Neural networks perform object detection and presented new methods for approaching object tracking, distance estimation, and speed prediction [13]. The speed predictions can be integrated into collision risk estimators, thereby helping drivers avoid accidents.

1.4. Artificial Intelligence

Many researchers have recognized Self-driving vehicle technology potential. The vehicle can use an ANN prediction model whose outputs can derive a control strategy to improve FE [14]. Artificial Neural Networks (ANN) can model and extract unseen features and relationships, which makes it reliable to model complex prediction problems and patterns. They are a very powerful tool to predict the future output of any system, which enables the utilization of different strategies to improve fuel economy and safety of autonomous vehicles.



(a)



(b)

Figure 9: (a)Artificial intelligence subsets (b) DNN performance

The subsection of AI is Machine learning, shown in Figure 9 (a), which is about extracting knowledge from data [15]. It is a research field at the intersection of statistics, artificial intelligence, and computer science and is also known as predictive analytics or statistical learning. Machine learning is really fast in terms of getting results. Another advantage of the machine learning algorithm is that they are easy to interpret and understand.

Deep Learning is a subset of machine learning, which has multilayered neurons. Deep Learning models are algorithms inspired by the structure and function of the brain called artificial neural networks. It Consists of hidden layers and multiple neurons. As we construct larger neural networks and train them with more and more data [16], their performance continues to increase, as shown in Figure 9 (b).

1.5. Literature

Many publications address the problem of velocity prediction. Mainly a prediction study should be able to address three main points. 1)Are there certain types of perception algorithms that work better than others (e.g., Markov chain, autoregression, machine learning)? 2)Are there specific sensor inputs that enable high-quality predictions (e.g., GPS, ADAS detections, V2I, weather information)? 3) Which is a better model with sensor/signal inputs for different prediction windows as an output? Starting in 2008, the earliest and presently most cited research about the prediction model comes from researchers at the University of Florida and the University of Wisconsin-Milwaukee. Their study used V2I and GPS signals as inputs into a perception model. The perception model was then able to output vehicle operation predictions using an analytical traffic model [17]. The approach was later revised to employ an artificial neural network (NN) [18]. The use of a NN to characterize a driving pattern was found to be effective.

In 2014, researchers at the University of Minnesota applied a traffic model to predict future vehicle velocity with V2V and V2I as inputs [19]. Prediction using ego vehicle velocity over 1-10 sec was compared for parametric and non-parametric models in Lefèvre et al. [20]. Results show that simple models performed well for short term prediction and advanced models for long term prediction.

Then, in 2015, researchers from the University of California at Berkeley recognized the critical relationship between perception and planning and investigated three perception models for use with a Model Predictive Control Optimal EMS. They use previous driving data and the current vehicle state as inputs prediction models. Their prediction model includes an exponentially varying perception model, a stochastic Markov chain perception model, and an NN perception model [21]. Additional studies by this group have shown FE results closer to optimal when traffic information is included [22]. A deep learning network is also used to predict ego vehicle velocity and route in Lemieux et al. [23]. Amir Rezaei et.al. Studied prediction for 1, 6, 10 seconds with GPS/GIS used in ANN [24]. Hellström and Jankovic proposed a model for a human driver operating an accelerator pedal and used it for prediction [25].

Starting in 2017, researchers at Colorado State University began publishing research that included a perception model, Optimal EMS, and FE results. The first study used current and previous vehicle velocity and GPS data input to a shallow NN perception model. They also developed an Optimal EMS computed using DP on a validated model of a 2010 Toyota Prius, which uses prediction from shallow NN. The maximum FE improvement was achieved using 30 seconds of prediction [26]. Deep Neural Networks (DNNs) is used to predict in Olabiyi et al. [27]. Zhang et al. utilized V2V and

V2I communications for future vehicle velocity prediction [28]. They also developed an energy management strategy based on vehicle velocity prediction.

Researchers from the Beijing Institute of Technology, China, demonstrated more consistent FE improvements with the aid of historical velocity data and a Gaussian NN perception model [29]. Additionally, researchers from the Beijing Institute of Technology used previous drive cycle data as an input into a Markov chain and NN perception subsystem with an Optimal EMS based on a genetic algorithm. Also, David Baker et al. and researchers at Colorado state university studied different prediction windows for error distribution with the NARX model [30]. They used Vehicle speed and GPS from CAN in the NARX model.

Recent trends have begun to implement advancements in machine learning and have replaced shallow NNs with deep NNs. Research conducted at the University of Michigan explored a variety of perception models. They studied auto-regressive moving average, shallow NN, long short-term memory (LSTM) deep NN, Markov chain, and conditional linear Gaussian models for prediction accuracy. Their study concluded that the LSTM deep NN provided the best prediction fidelity (measured in mean absolute error) [31]. Furthermore, using the prediction from LSTM, they realized a fuel economy improvement of 3% in the dynamometer validated model of a 2017 Toyota Prius Prime [14].

1.6. Novel Contribution

Despite the fact that the issue of vehicle speed prediction has been previously studied, the dependency of prediction on different groups of signals still needs to be understood. The prediction model study should address research gaps in the project. Such as, are there certain types of perception algorithms that work better than others? Are there specific sensor inputs that enable high-quality predictions? Which is a better model with sensor/ signal inputs for different prediction windows as an output?

Considering the huge potential in the above applications and implementations, we used various methods, which include ANN and machine, learning models. We gathered numerous different signals along the roads of Fort Collins, Colorado, to get the custom drive cycle for the velocity prediction. Moreover, the research expands into the effect of these signals and models on different prediction horizons. For a better assessment, we used two different assessment methods, which are MAE and time shift.

1.7 Thesis Overview

The research is organized as follows: In section 2, the vehicle velocity prediction strategy and drive cycle development are described. It also elaborates prediction models used for the project. Section 3 presents the assessment methods and the results in different scenarios. Finally, section 4 summarizes the conclusions and sets the direction of discusses future research in terms of implementation.

2. METHODOLOGY

2.1 Vehicle Velocity Prediction Strategy

The objectives of this research are to assess the potential for radar signal, GPS, EGO vehicle parameters, and V2I data to improve vehicle velocity prediction. We are also focusing on the effect of these signals and the effect of different ANN models on different prediction windows. In the strategies discussed in section 1.3, an inaccurate prediction can deteriorate energy savings or even cause safety concerns. Because of such dependence on the prediction horizon, there is a pressing need to develop accurate and robust approaches to predict vehicle velocity for getting better results.

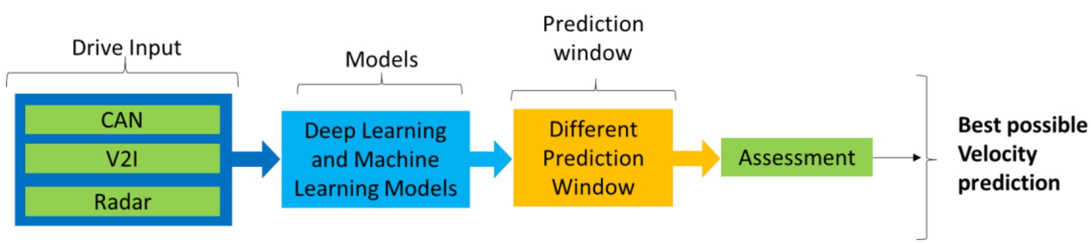
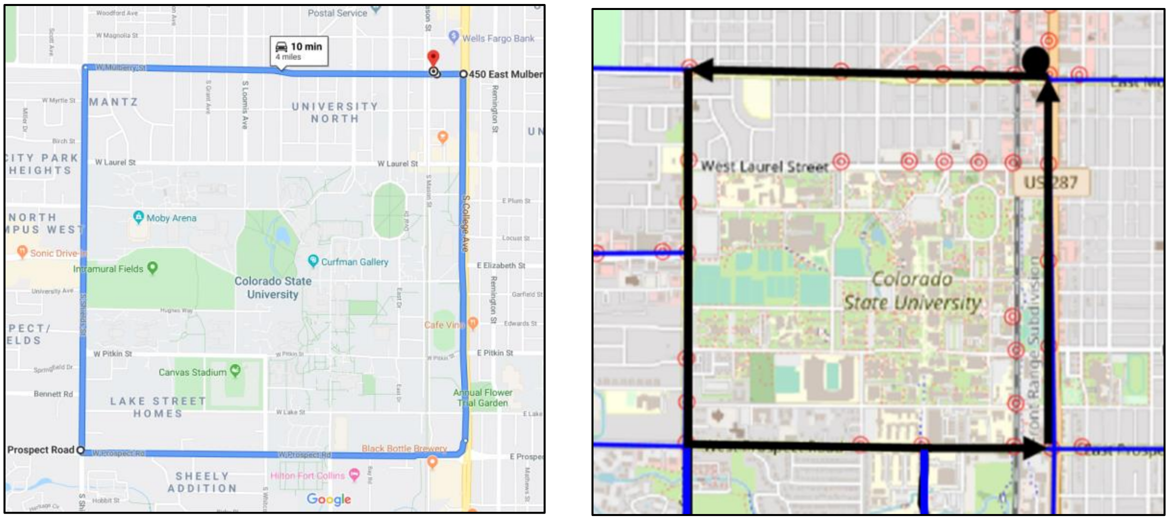


Figure 10: Vehicle velocity prediction strategy

Figure 10 represents the overall strategy used in the research. First, a sensor-equipped vehicle collects drive inputs by driving along the route. Developed deep neural networks and machine learning models uses these collected datasets as input to predict the velocity. The prediction model can give output for different prediction windows. These results are then assessed to find out the best possible velocity prediction.

2.2. Drive Cycle Development and Signal Recording

To obtain signals for drive input, a driving vehicle equipped with different sensors such as GPS, CAN data logger, and radar is driven on the route in Fort Collins, Colorado.



(a) (b)

Figure 11: (a)Drive cycle map of the Fort Collins dataset (b) traffic signal map

Figure 11(a) shows the view of the GPS map. Traffic signal map is shown in Figure 11(b), with the black arrows representing the driving directions, the black circle representing the start point and endpoint, the red targets representing traffic signals, and the blue lines representing traffic segments. In total, this route is 4 miles and should take 10-12 minutes per cycle, depending on traffic. The Fort Collins dataset was collected in October 2019 and contained data from repeated drives along a fixed route by the same driver. This route represents the round trip on the following:

1. Parking Lot
2. West on Mulberry until Shields
3. South on Shields until Prospect
4. East on Prospect until College
5. North on College until Mulberry
6. West on Mulberry until Parking Lot
7. Parking Lot

The test generated Autonomous Driver Assistance System (ADAS) data for the vehicle forward cone from smart radar, which is a standard setup for production vehicle ADAS systems. Test also generated the Vehicle to Infrastructure (V2I) data in the form of traffic signal information and segment travel times. The ego vehicle was specially instrumented research vehicle with the afore-mentioned ADAS sensors and a stereo camera with a Freematics logger. It is important to note that the timestep used for the analysis is 0.1 seconds.

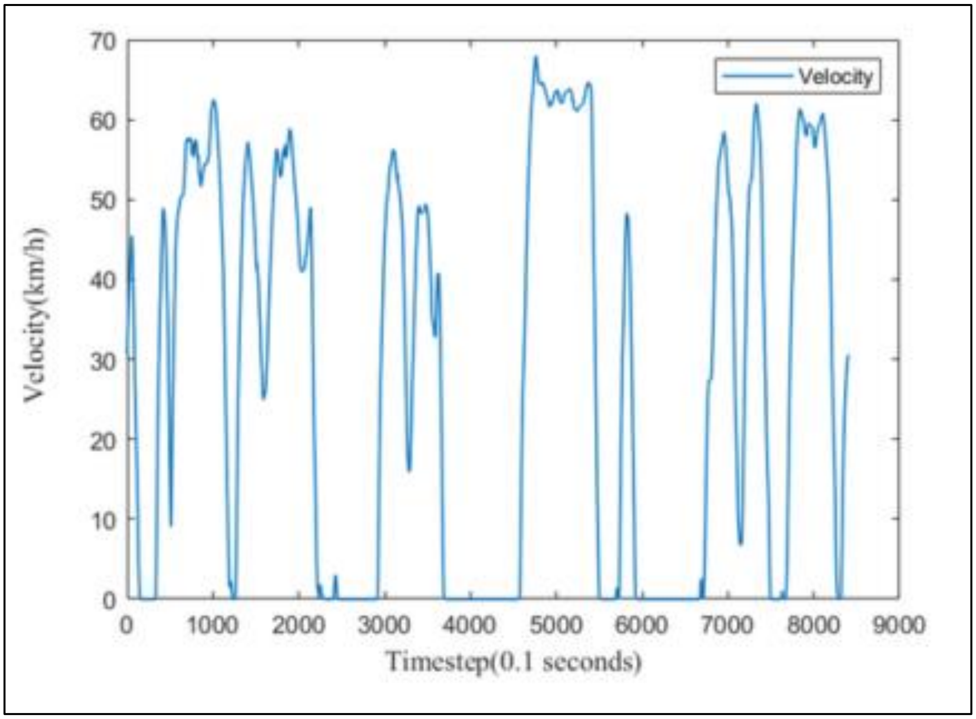


Figure 12: Velocity versus time for one drive instance

Figure 12 shows the data plot for Velocity vs. time for one drive instance. Figure 13 shows the plotted longitude versus latitude versus time. The velocity is on the map against time for three drive instances. We can observe that the velocity profiles are a similarity.

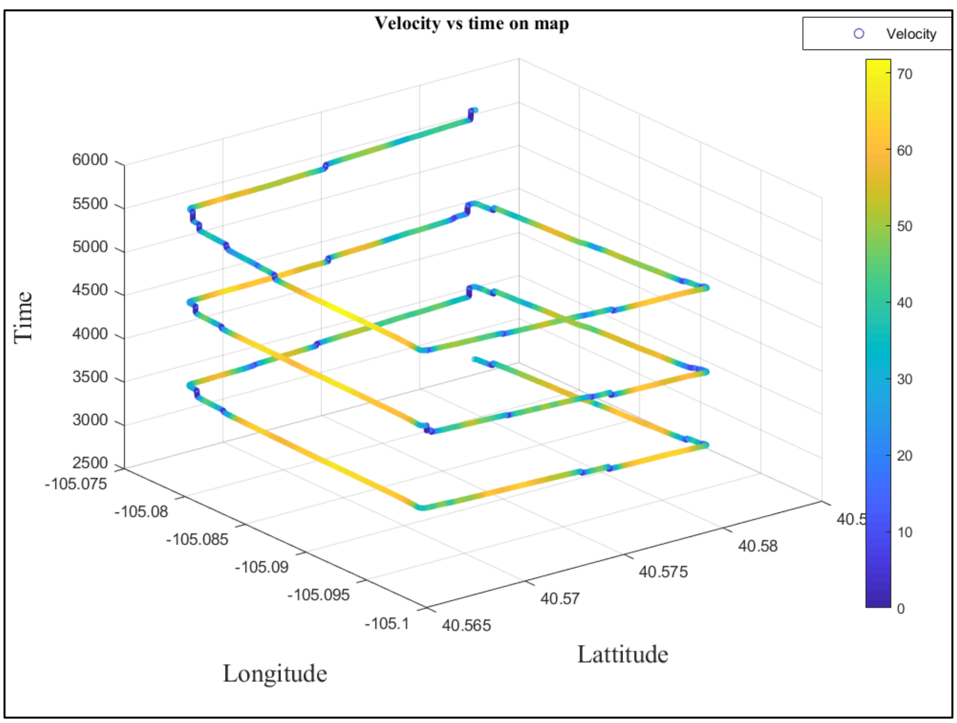
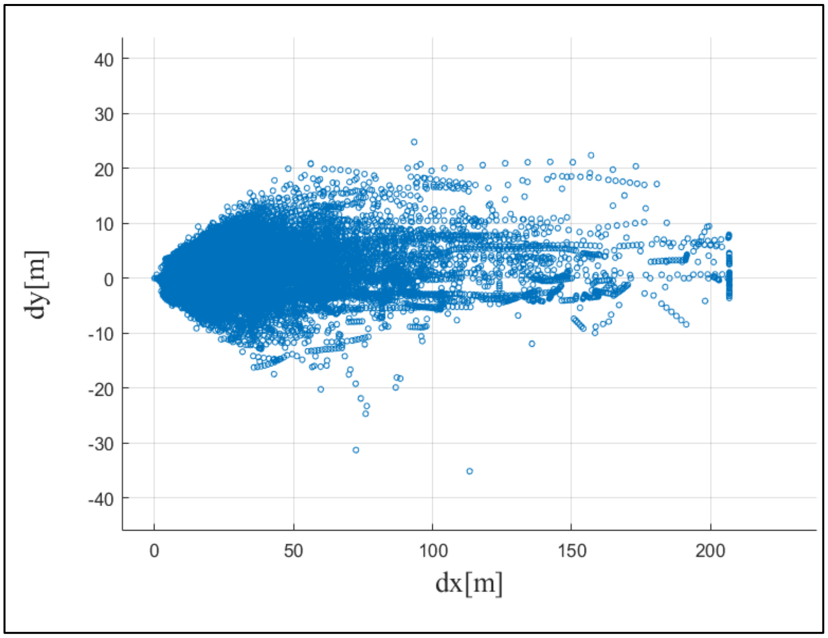
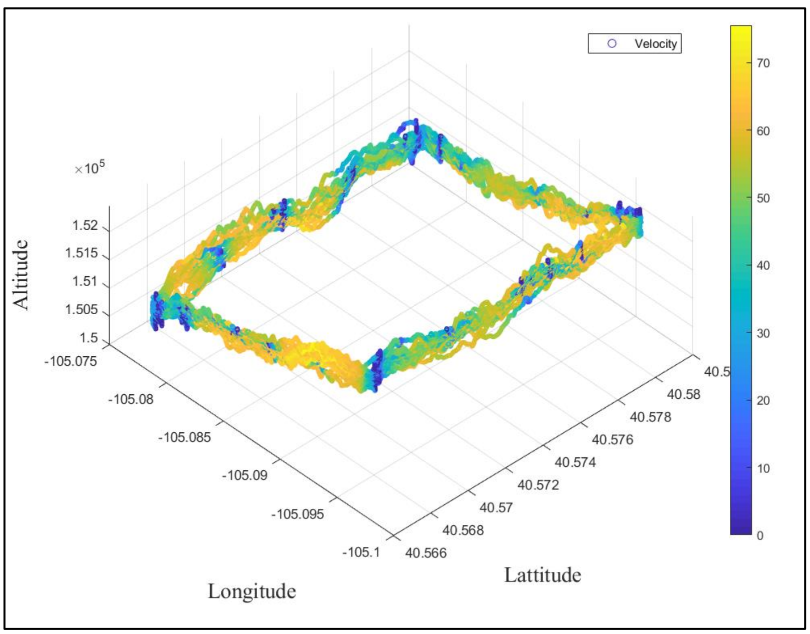


Figure 13: Velocity on the longitude versus latitude versus time



(a)



(b)

Figure 14: (a)Radar object detections immediately in front (b)Velocity on map terrain

The radar objects detections immediately in front are plotted and shown in Figure 14(a). These show the objects detected and position from the vehicle. Figure 14(b). Shows the plot for the longitude versus latitude versus altitude, i.e., velocity, is shown on this map terrain.

Table 1 shows all collected signals on the drive cycle. The signals in the red box represent the data collected from the EGO vehicle, which consists of the ego vehicle parameter and the ego vehicle's position (GPS). Signals in the blue box represent V2I data, which includes the Traffic Signal Phase and Timing (SPaT), Segment speed (SS), and radar data.

Table 1: List of collected signals in drive input

Sr. No.	Parameters	Sr. No.	Parameters
1	Time	11	Turn Signal
2	Latitude	12	Vehicle Speed
3	Longitude	13	Acceleration Longitude
4	Distance	14	Acceleration Latitude
5	Brake Pedal Position	15	Yaw Rate
6	Transmission Gear	16	Altitude
7	Engine Speed	17	SPaT
8	Max Torque	18	Segment Speed
9	Min Torque	19	ADAS
10	Engine Torque		

All these signals are grouped in different combinations, which are used as input to the neural network and machine learning models. This is discussed in the subsequent section.

2.3 Neural Networks and Machine Learning Models

2.3.1 Artificial Neural Networks and Machine learning analogy

Artificial Neural Networks are algorithms consist of multilayered neurons that are inspired by the structure and function of the brain. The human brain can be considered into four different sections, depending on their functions are shown in Figure 15. A trained neural network is generally the one with stored weights. They can act as long-term memory or store data through a time similar to the frontal lobe. For object detection convolutional neural networks (CNNs) are used, which acts as an occipital lobe whose primary function is the vision. CNN could also perform feature extraction and regression model. Current technology is limited in terms of the parietal lobe, whose function is sensation, perception, and creating a spatial coordinate system. Recurrent Neural Networks (RNNs) can function as short term memory, which is similar to the frontal lobe, which is responsible for personality, behavior, and short-term memory. LSTM is a type of RNN, which is an excellent example of short term memory. ANN offer increased flexibility and can scale in proportion to the amount of training data available.

On the other hand, deep learning neural networks are nonlinear methods. Machine learning Algorithms are easier to understand and more comfortable to implement. Classical machine learning algorithms are also very interpretable, which means it is much easier to understand the reasoning for the model's prediction.

Subsequent sections explain the different models used in the project. The lit review shows the use of DNN in different prediction models. It was observed that LSTM could perform better for prediction. In many cases, it can be observed that CNN can perform better than LSTM. We are also using different machine learning for their interpretability.

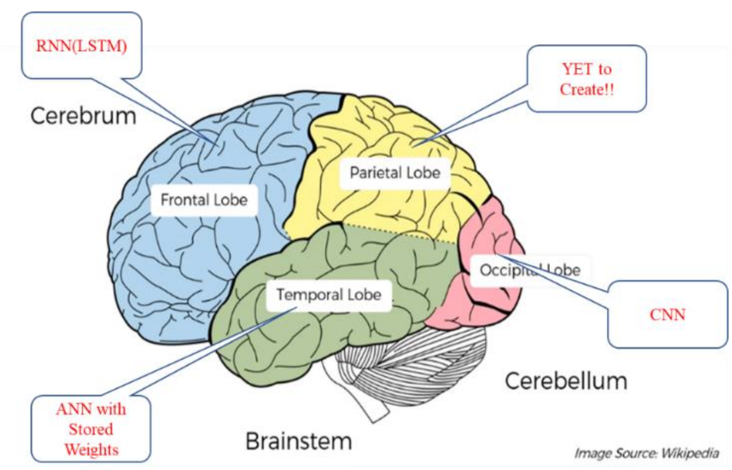


Figure 15: ANN analogy

2.3.2 Neural Network and Machine learning Algorithms:

The neural networks and machine learning models used for the study are shown in Table 2.

Table 2: Neural networks and machine learning models

Type	**Network**
RNN	LSTM
	CNN LSTM
CNN	CNN
Machine Learning	Decision Trees
	Bagged Trees
	Random Forest
	Extra Trees
	Linear Regression
	LR With Interactions
	Ridge
	KNN

a) LSTM (Long-Short Term Memory):

LSTM is a special type of Recurrent Neural Network (RNN) capable of learning long-term dependencies from large data sets (i.e., data sets with lots of inputs) and avoiding issues such as gradient vanishing/exploding of conventional RNNs. For the analysis of sequential data, LSTM is the most commonly used deep learning model. An RNN composed of LSTM units is often called the LSTM network, where units of RNN are LSTM units.

Inside the repeating parts, the conventional RNNs have only one activation function, while LSTMs have three gates with different activation functions, interacting with each other. A typical LSTM unit, often called a memory cell, memory block, is composed of a cell, an input gate, an output gate, and a forget gate [32]. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell. The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. C_{t-1} and C_t are old and new cell state, respectively. h_{t-1} and h_t are outputs of the previous and current cell. X_t is the input to the cell.

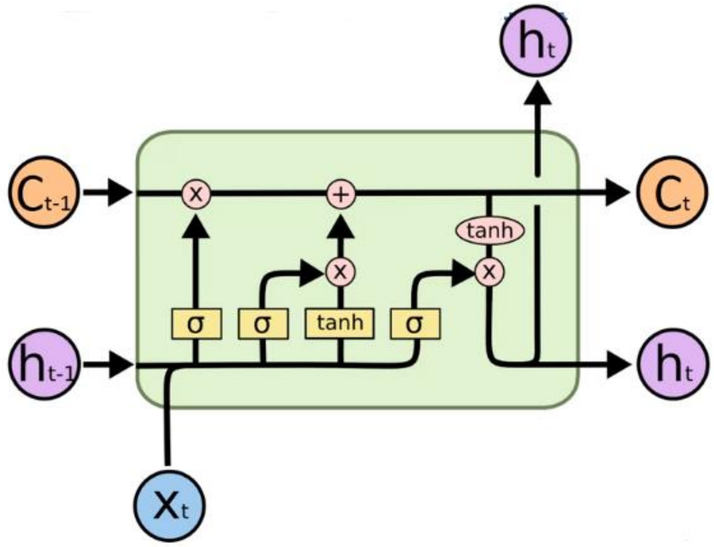


Figure 16: LSTM cell

The typical LSTM Cell internal structure shows in Figure 16. Computation details can be found in [33-35]. The gates are computation can be summarized by the following equations.

❶ Forget Gate: Decides what information to discard from the cell.

$$f_t = \sigma\left(W_f[h_{t-1}^T, X_t^T]^T + b_f\right)$$

❷ Input Gate: Decides which values from the input to update the memory state.

$$i_t = \sigma(W_i[h_{t-1}^T, X_t^T]^T + b_i)$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}^T, X_t^T]^T + b_C)$$

❸ Output Gate: Decides what to output based on input and the memory of the cell.

$$o_t = \sigma(W_o[h_{t-1}^T, X_t^T]^T + b_o)$$

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t$$

$$h_t = o_t \times \tanh(C_t)$$

where W_f, W_i, W_C, W_o and b_f, b_i, b_C, b_o designate vectors (matrices) of weights and biases for forget gate, input gate, cell state, and output gate, respectively. In the above expressions, σ is a sigmoid function.

b) CNN (Convolutional Neural Networks):

Convolutional Neural Networks are a powerful artificial neural network technique. These networks preserve the structure of the problem and used for object recognition such as handwritten digit recognition. They are popular because people are achieving state-of-the-art results on challenging computer vision and natural language processing tasks. They are also used for regression problems and sequence prediction. In many cases, CNN can outperform LSTM models. CNN can achieve what LSTM has been used, and performs better in predicting sequences, but in a much faster, more computationally efficient manner [36-37].

There are three types of layers in Convolutional Neural Network shown in Figure 17:

1. Convolution Layers
2. Pooling Layers.
3. Fully Connected Layers.

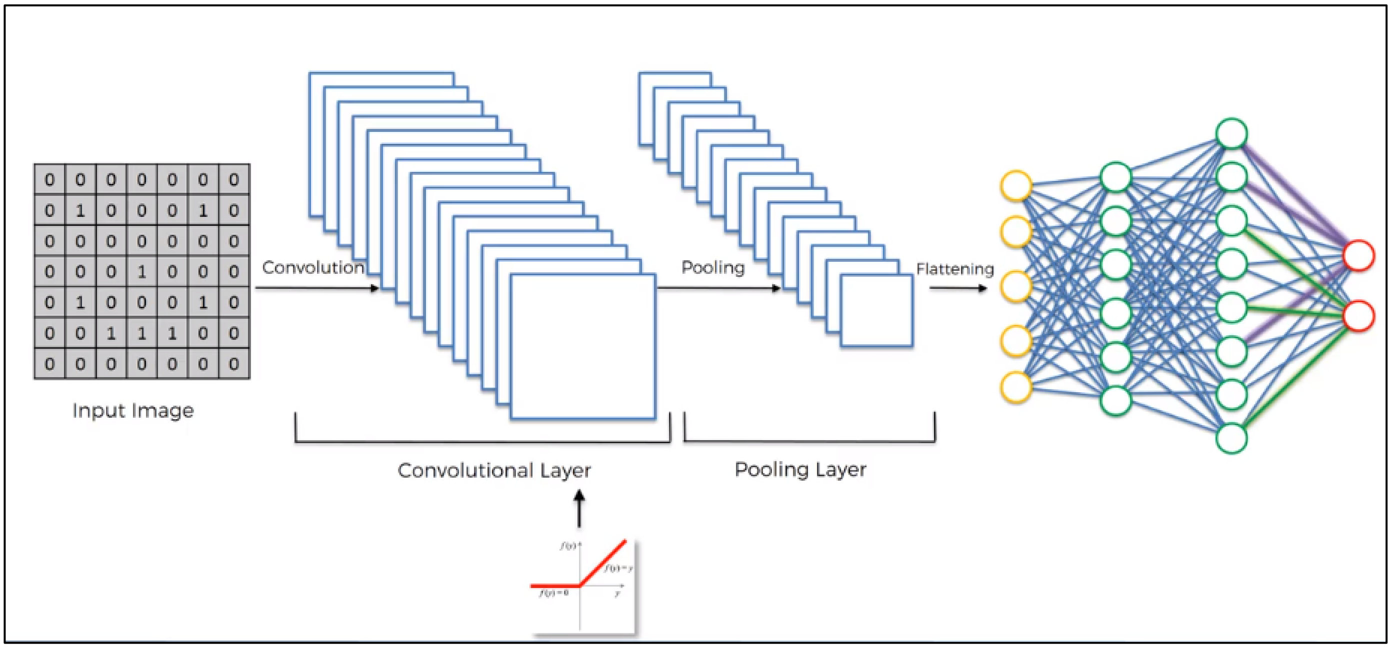


Figure 17: CNN structure

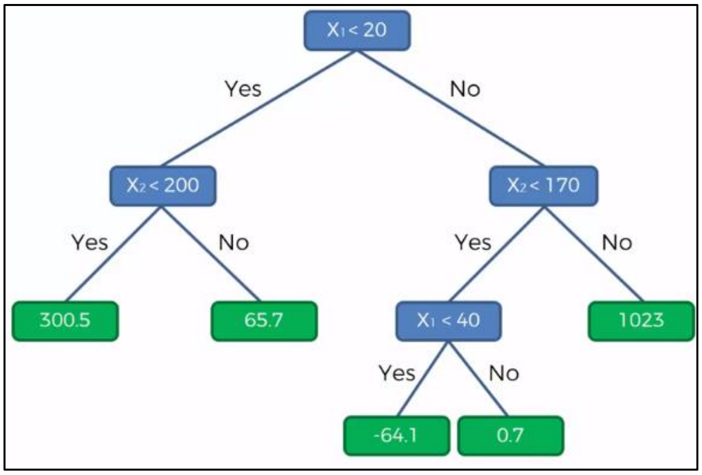
Convolutional Layers are comprised of filters and feature maps. Features have both weighted inputs and generate an output value like a neuron. The feature map is the output of one filter applied to the previous layer. A feature map can also be called the collected output of the filters.

The pooling layers down-sample the previous layers feature map. Pooling layers follow a sequence of one or more convolutional layers and are intended to consolidate the features learned and expressed in the previous layers feature map. Pooling can be considered a technique to compress or generalize feature representations and generally reduce the overfitting of the training data by the model.

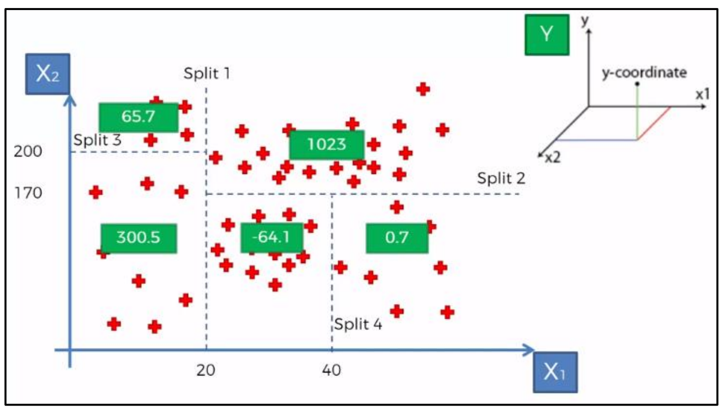
Fully connected layers are the standard flat feedforward neural network layer. Fully connected layers are used at the end of the network after the convolutional and pooling layers have performed feature extraction and consolidation. They are used to create final nonlinear combinations of features and for making predictions by the network.

c) Decision Tree

Decision Tree algorithms are used for classification and regression predictive modeling problems [38]. Each node represents a single input variable (x) and a split point on that variable. The leaf nodes of the tree contain an output variable (y), which is used to make a prediction. Given a dataset with two inputs of X1 and X2, the output of y, below Figure 18(a), is an example of a binary decision tree.



(a)



(b)

Figure 18: (a)Example decision tree (b) partitioning of input space

Making predictions is relatively straightforward. Given a new input, the tree moves forward by evaluating the specific input starting at the root node of the tree. A learned decision tree is a partitioning of the input space shown in Figure 18(b). The input variable can be considered as a dimension on a p-dimensional space. The decision tree split this up into rectangles (when p = 2 input variables) or hyper-rectangles with more inputs. New data is filtered through the tree and lands in one of the rectangles, and the output value for that rectangle is the prediction made by the model.

d) Ensembles (Random Forest, Bagged Tree, Extra Tree):

Ensemble learning is a technique that combines the predictions from multiple neural network models to reduce the variance of predictions and reduce generalization error [39]. Bootstrap Aggregation (or Bagging) involves taking multiple samples from the training dataset and training a model for each sample. The final output prediction is averaged across the predictions of all of the sub-models. The three-bagging model used in the research are:

- Bagged Tree
- Random Forest
- Extra Trees

Bagging (Bootstrap Aggregation) is a procedure that can be used to reduce the variance for those algorithms that have high variance, typically decision trees. Decision trees are sensitive to the specific data on which they are trained. If the training data is changed, the resulting decision tree can be quite different, and in turn, the predictions can be quite different. In bagged trees, many random subsamples of the input dataset are created. Decision trees are trained on each of these subsamples. When the new dataset is given as input, the average prediction from each model is calculated.

Random Forests are an improvement over bagged decision trees. Decision trees tend to get greedy, meaning they make an optimal local choice at each node. They choose which variable to split on using a greedy algorithm that minimizes error. Even with bagging, the trees can have a lot of structural similarities, and it turns to result in a high correlation in their predictions. Combining prediction from multiple models in the ensemble's works better if the predictions from the sub-models are not correlated or weakly correlated. The random forest changes the algorithm for the way that the sub-trees are learned so that the resulting predictions from all the subtrees have less correlation. When selecting a split point in decision trees, the learning algorithm can consider through all variables and all variable values to select the most optimal split point. The random forest algorithm changes this procedure so that the learning algorithm is limited to a random sample of features.

Extra Trees (Extremely randomized Trees) is a type of ensembles where random trees are constructed from samples of the training dataset. Extra trees randomize certain decisions and subsets of data to minimize over-learning from the data and overfitting [40]. Its two main differences with other ensembles methods are that it splits nodes by choosing cut points entirely at random. This leads to more diversified trees and fewer splitters to evaluate when training an extremely random forest. Also, Extra Trees seem to keep higher performance in the presence of noise features.

e) Linear Regression:

Linear regression is one of the most common techniques used for regression analysis. The linear regression model can also be referred to as estimating the values of the coefficient (weights and bias) used in the representation training data [39]. When we have multiple input features, it is called multiple linear regression. In that case, linear regression uses ordinary least squares to estimate the values of coefficients. The ordinary least square procedure seeks to minimize the sum of squared residuals. Considering a regression line, the distance from each data point from that line is squared, and the sum of all the squared errors, which is then minimized. Linear regression uses gradient descent for iteratively minimizing the squared error. In linear regression, interaction terms can be taken into account. Considering the interaction terms between the features can significantly expand understanding of the relationships among features in the model and can improve the prediction.

Ridge Regression is an extension of linear regression where the loss function is modified to minimize the complexity of the model measured as the sum squared value of the coefficient values. Ridge regression learns weights and bias, using the least-squares criterion but adds a penalty for large variations in weights parameters. Once the parameters are learned, the ridge regression uses least squares for prediction. The addition of a parameter penalty is called regularization. Regularization prevents overfitting by restricting the model, typically to reduce its complexity.

f) KNN (K- Nearest Neighbor):

K-Nearest Neighbor is the algorithm used for classification and regression, which uses training data directly to make predictions [39]. Predictions are made for a new data point by searching through the entire training set for the ‘k’ most similar instances(neighbors) and averaging the output variable for those k instances.

2.4. Training and Testing

Recorded signals consist of 13 drive instances. These recorded data is split between training data consisting of 12 drive cycles and testing data consisting of 1 drive cycle, as shown in Figure 19.

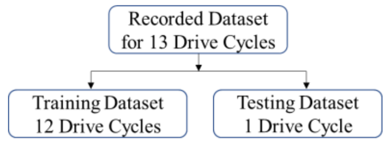


Figure 19: Training and testing dataset split

The training data is given as an input to the developed models. Then the testing dataset is given as input to the trained network to get the predicted output. This is shown in Figure 20

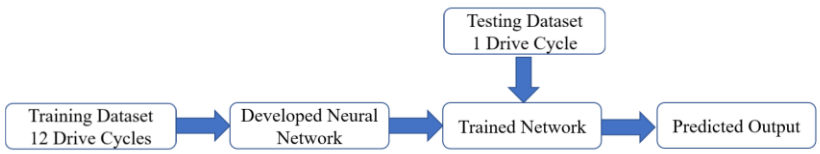


Figure 20: Training and testing input to neural network

Implementation of Code is shown below:

Create training and testing datasets in the MATLAB, such that, training dataset contains 12 drive instances and testing dataset contains 1 drive instance.

Import the training and testing dataset in python environment

Import required model library from the keras/scikit learn.

Reshape the input data according to requirement of model.

Create model and give the training data to the model.

Then give testing data to the model and get predicated velocity.

Tune the model to get accurate predictions.

Save final output to MATLAB file.

The results then can be analyzed based on assessment methods using MATLAB toolbox.

Above method is repeated for each drive instance to find the prediction for that instance. This way we are able to calculate the cross-validation results for velocity prediction.

3.RESULTS

3.1. Assessment

For accurate assessment of the prediction results, we introduced two assessment criteria: Mean absolute error (MAE) and Time shift.

a) Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is a measure of the difference between the two variables. Assume $y_{\mathrm{t}1}, \ldots, y_{tn}$ are prediction results and $Z_{\mathrm{t}1}, \ldots, Z_{tn}$ are target values. The MAE is given by,

$$\mathrm{MAE}(\mathrm{Y_t}, \mathrm{Z_t}) = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} |\mathrm{y_{ti}} - \mathrm{Z_{ti}}|}{\mathrm{n}}$$

Smaller MAE means smaller errors between prediction results and target values.

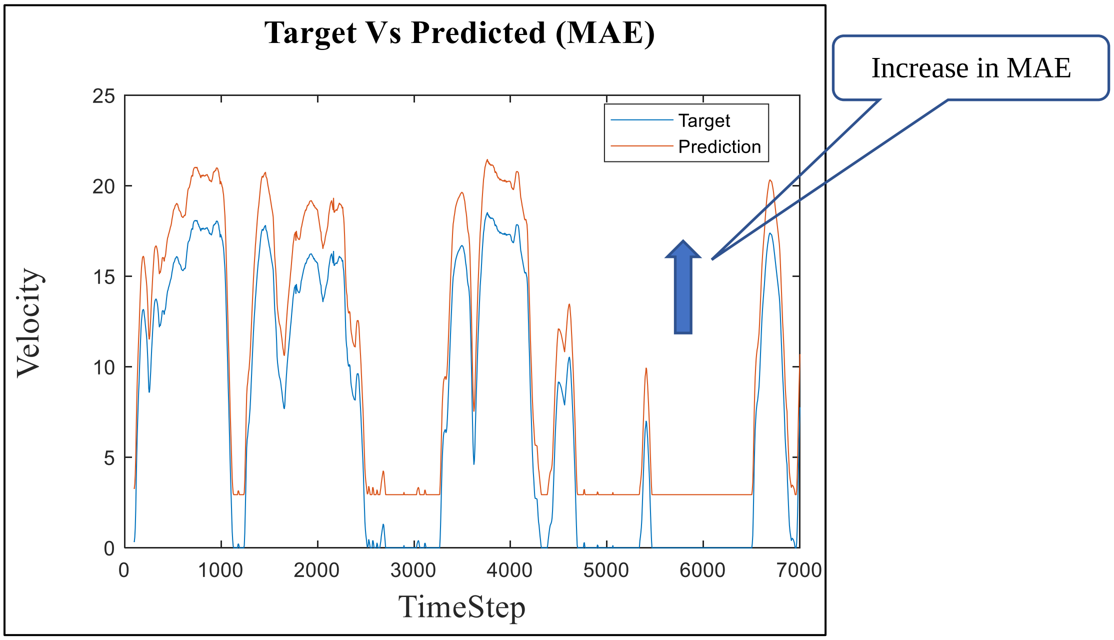
b) Time Shift

The time shift is a measure of the time lag between the predicted time series and target time series. It uses a cross-correlation technique for finding the time shift error. A smaller time shift means a smaller time lag between the prediction results and target values. It can be expressed as,

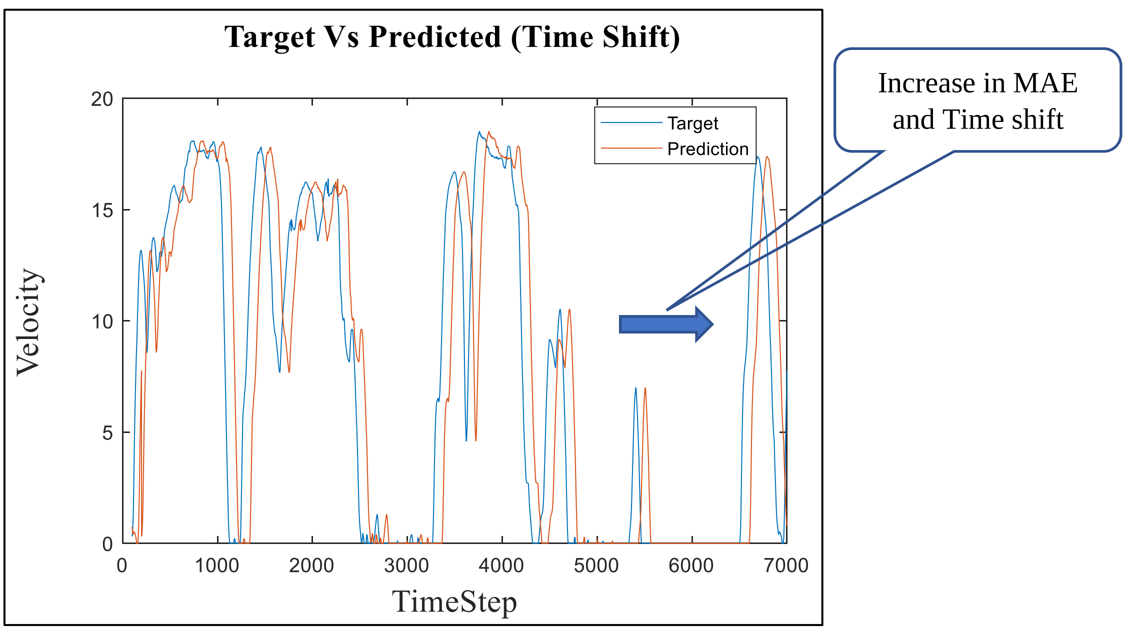
$$\text{time shift } = \mathrm{arg}_{\delta}\mathrm{max}\,(\sum_{0}^{n} |\mathrm{Y}_{\mathrm{t}-\delta} * \mathrm{Z}_{\mathrm{t}1}|)$$

3.1.2. Analysis of MAE and Time Shift

The effect of accuracy of MAE and time shift on fuel economy and safety improvement strategy is unknown. Though to understand the assessment methods, we consider two cases of different hypothetical predictions are shown in Figure 21. The red line represents target velocity, and the blue line represents the prediction. It is important to note that, MAE in both cases is 1.6763 m/s. In Figure 21(a), we can observe the upward shift in prediction, while in Figure 21(b), we can see a sideways shift. In the first case, the MAE increases as upward shift increases while the time shift remains the same. In the second case, an increase in MAE and time-shift both is observed with an increase in time shift error up to 10 seconds. We can also see that prediction in the first case is not zero at any stop event. On the other hand, velocity predictions are correct in the second case, but it is predicted at the wrong time.



(a)



(b)

Figure 21: (a) Upward shift in prediction (b) Sideways shift in prediction

In the case of actual prediction, it is essential to understand that it may contain both types of errors. Figure 22 shows the example which has MAE of 4.3901 m/s and the time shift of 10 seconds.

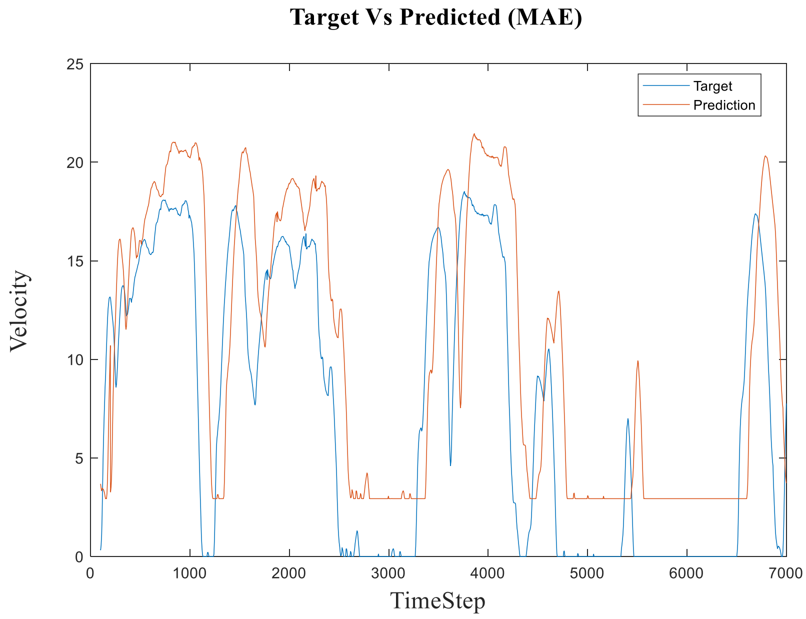


Figure 22:Hypothetical example of prediction consisting of both errors

Analysis of results.

For better understanding, we have divided the study into the following sections:

- Effect of **Different Signals** on prediction.
- Effect of **Different Models** on prediction.
- Effect on **prediction window** by different signals and different models.
- **1- 10 seconds** forward prediction.

3.2. Effect of Different Signals on Prediction

For comparing different signals, we divided the signals into different groups. The prediction plots show a comparison of the prediction of the 10th second using the LSTM model against the Target values.

Table 3: Signals used in group A

Data Group A	Signals
	Current Velocity
	GPS

Group A data contains current velocity and GPS, as mentioned in Table 3. Figure 23 shows the results for the velocity prediction for the 10th second using Group A signals and plotted against the target value. It can be observed that the prediction results are not aligning with the target.

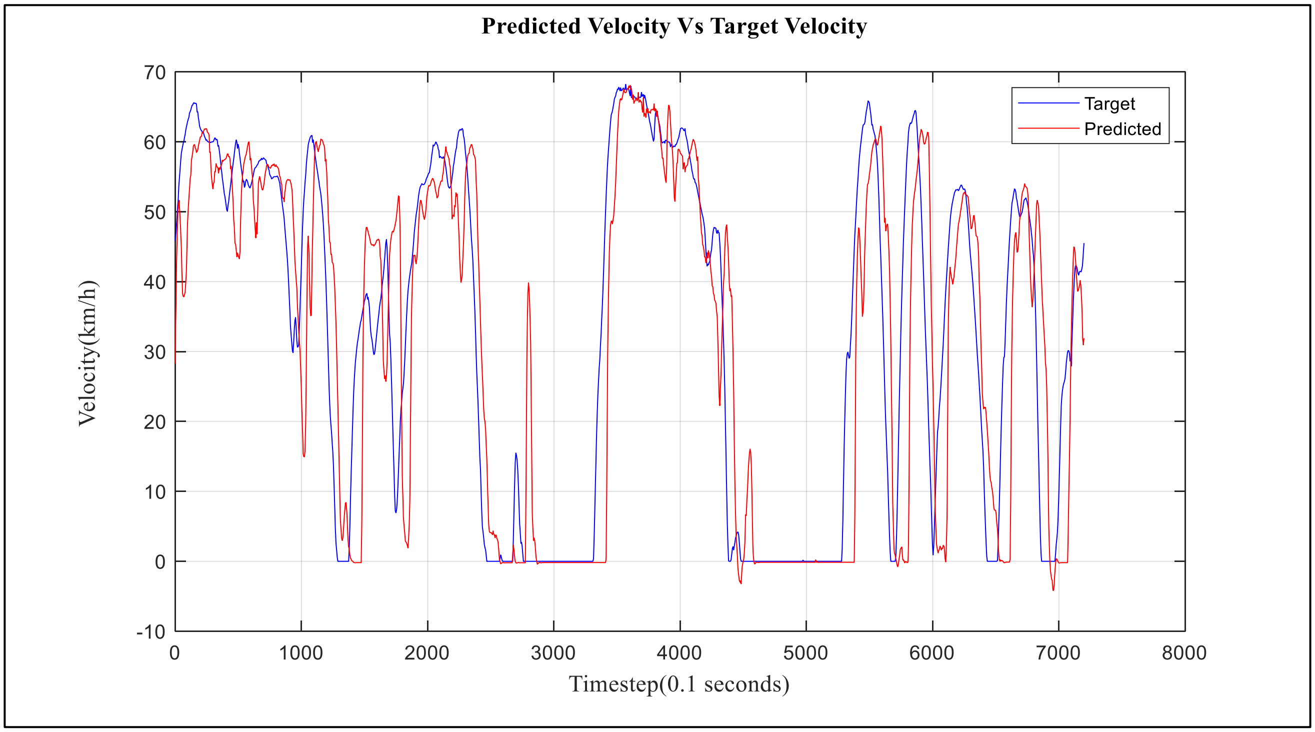


Figure 23: 10th-second prediction results using group A signals

Table 4: Signals used in group B

	Signals
Data Group B	Current Velocity
	GPS
	Previous 5 Seconds
	EGO and Engine Parameters

Signals
Group A
Previous 5 Seconds
EGO and Engine Parameters

Group B data contains current velocity, GPS, previous 5 seconds of velocity, and all EGO and Engine parameters, as shown in Table 4. Figure 24 Velocity prediction for 10th second using Group B signals and plotted against the target value.

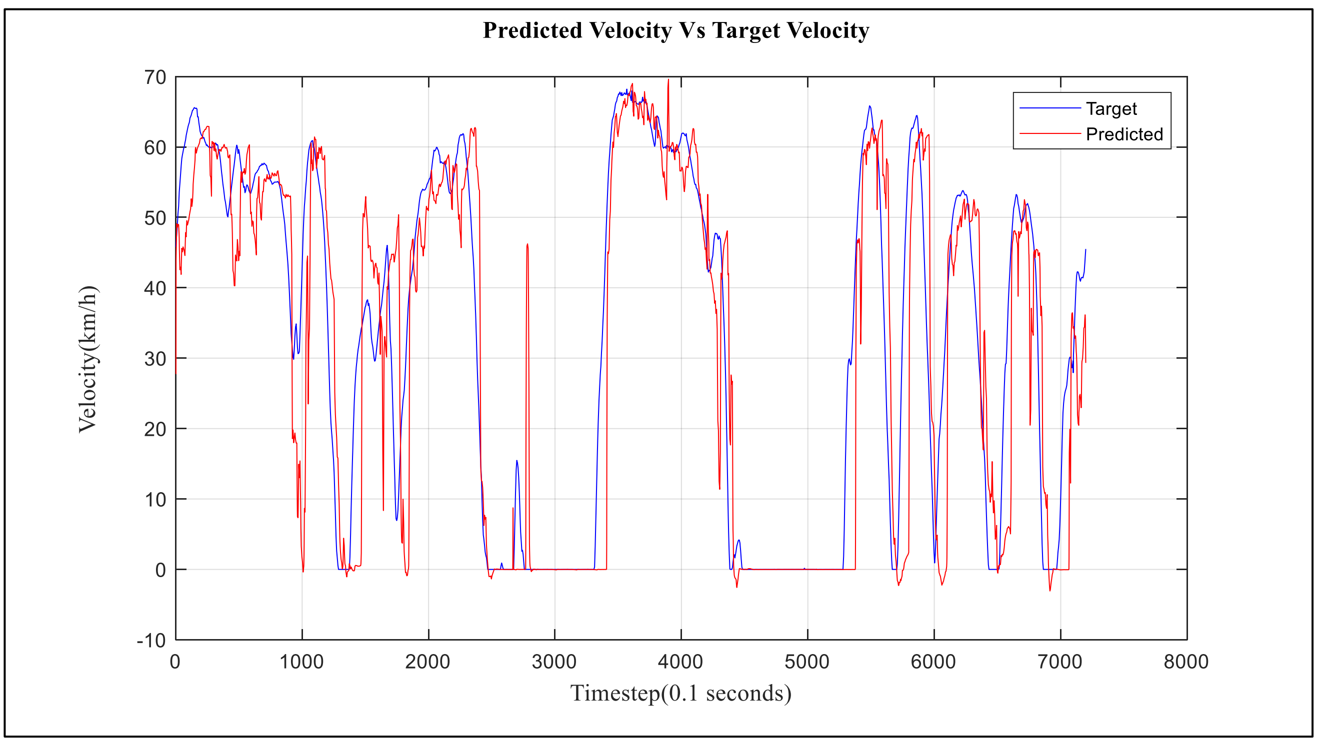


Figure 24: 10th-second prediction results using group B signals

Group C data contains current velocity, GPS, previous 5 seconds of velocity, EGO and Engine parameters, and radar data, as mentioned in Table 5. Figure 25 shows the velocity prediction for the 10th second using Group C signals and plotted against the target value.

Table 5: Signals used in group C

	Signals
Data Group C	Current Velocity
	GPS
	Previous 5 Seconds
	EGO and Engine Parameters
	Radar Data

⇔

Signals
Group B
Radar Data

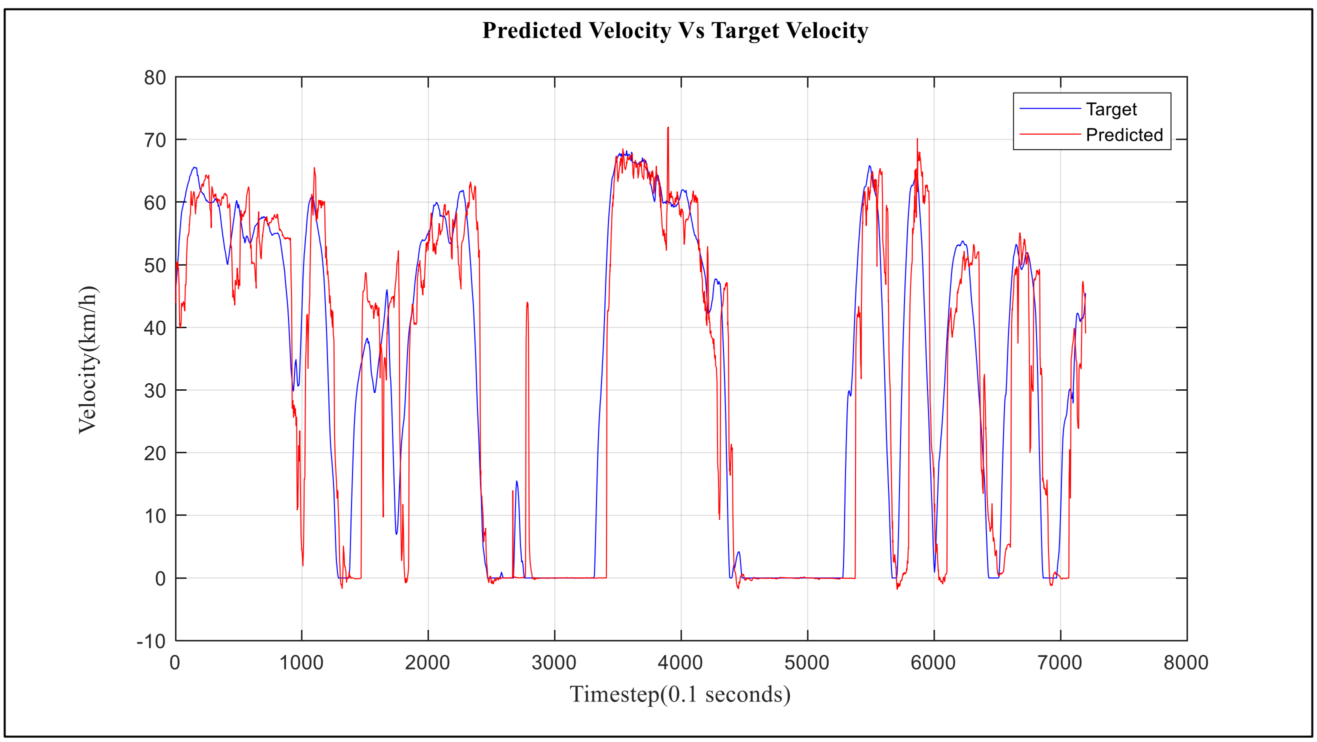


Figure 25: 10th-second prediction results using group C signals

Group D data contains current velocity, GPS, previous 5 seconds of velocity, EGO, and Engine parameters, and Spat (Signal Phase and Timing), as shown in Table 6. Figure 26 shows the velocity prediction for the 10th second using Group D signals and plotted against the target value. We can observe improvement in the alignment of the plots.

Table 6: Signals used in group D

	Signals
Data Group D	Current Velocity
	GPS
	Previous 5 Seconds
	EGO and Engine Parameters
	SPat

Signals
Group B
SPat

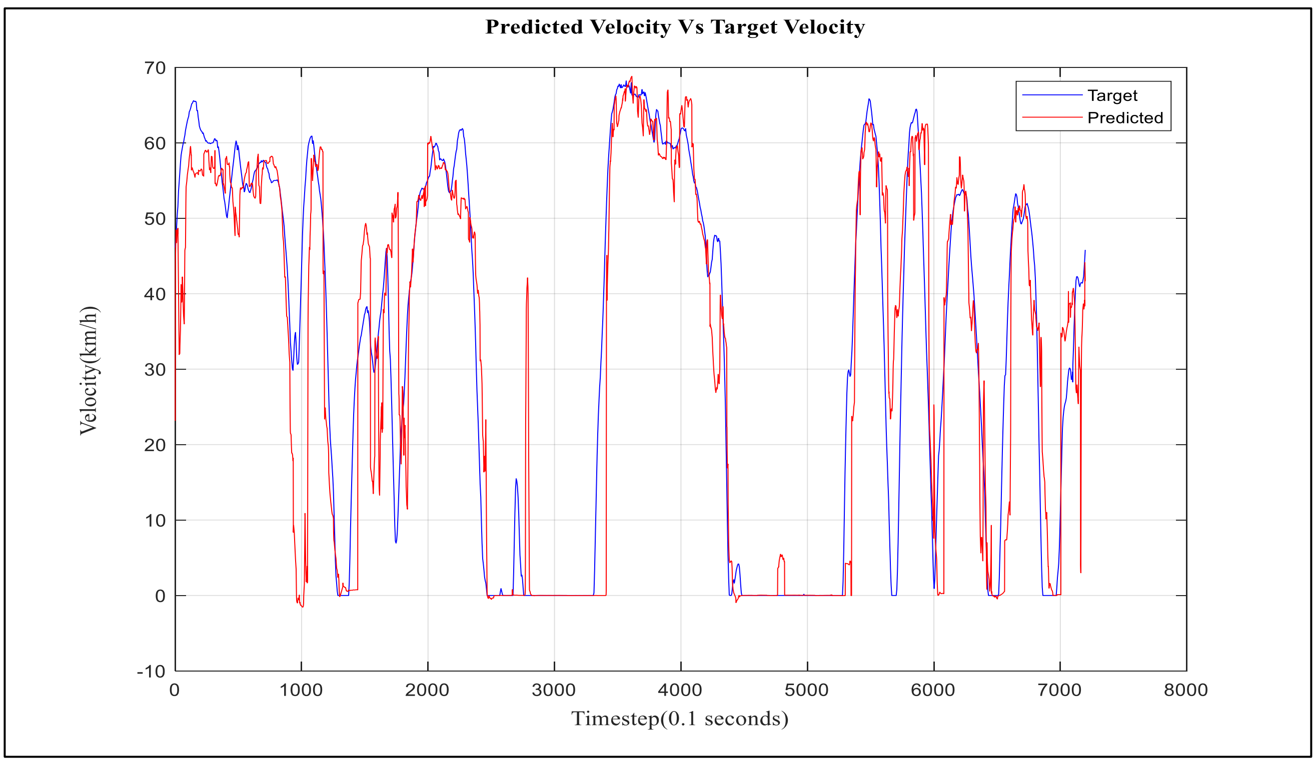


Figure 26: 10th-second prediction results using group D signals

Group E data contains current velocity, GPS, previous 5 seconds of velocity, EGO, and Engine parameters, SPaT (Signal Phase and Timing), and Segment speed, as shown in Table 7. Figure 27 shows the velocity prediction for the 10th second using Group E signals and plotted against the target value.

Table 7: Signals used in group E

	Signals
Data Group E	Current Velocity
	GPS
	Previous 5 Seconds
	EGO and Engine Parameters
	SPat
	Segment Speed

Signals
Group D
Segment Speed

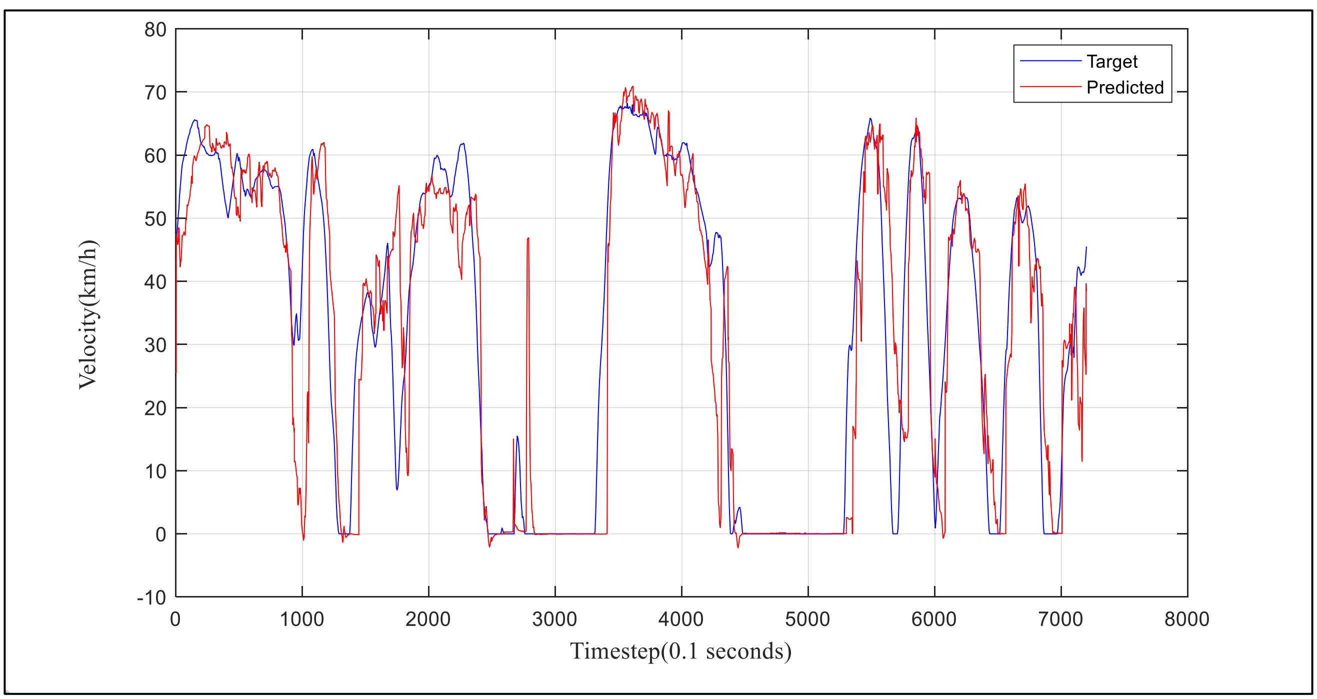


Figure 27: 10th-second prediction results using group E signals

Group F data contains current velocity, GPS, previous 5 seconds of velocity, EGO, and Engine parameters, SPaT (Signal Phase and Timing), Radar, and Segment speed, as mentioned in Table 8. Figure 28 shows the velocity prediction for the 10th second using Group F signals and plotted against the target value.

Table 8: Signals used in group F

	Signals
Data Group F	Current Velocity
	GPS
	Previous 5 Seconds
	EGO and Engine Parameters
	SPat
	Segment Speed
	Radar

Signals
Group E
Radar

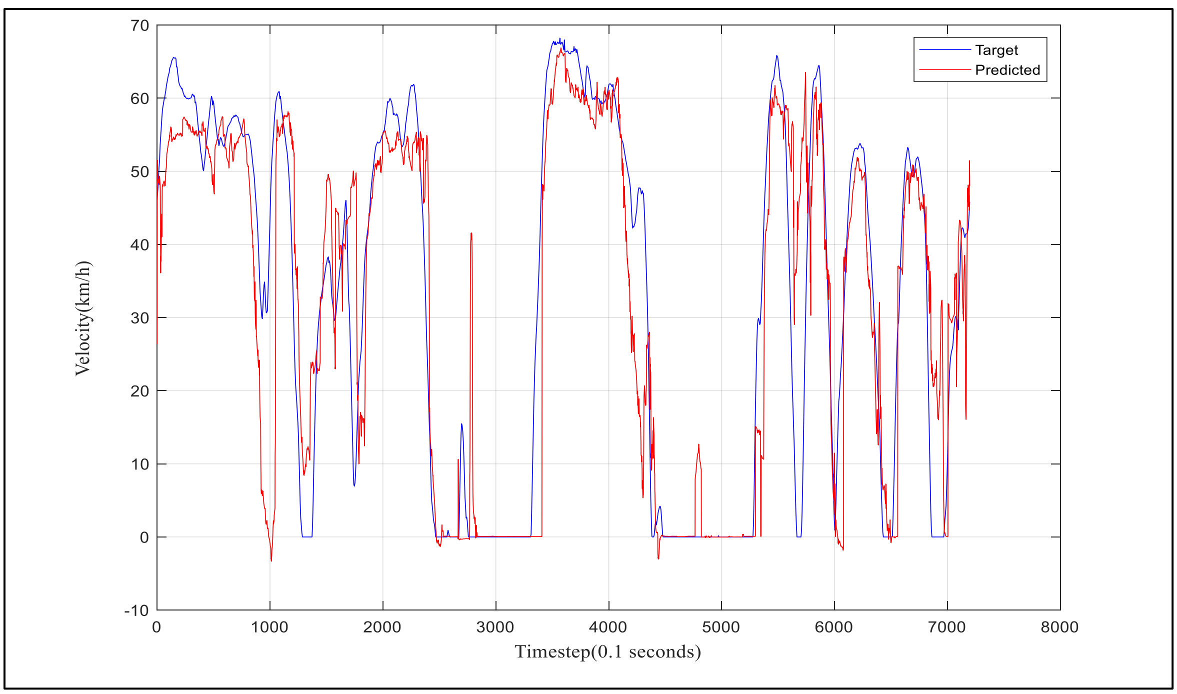


Figure 28: 10th-second prediction results using group F signals

Group G data contains current velocity, GPS, previous 5 seconds of velocity, EGO, and Engine parameters, and Segment speed, as shown in Table 9. Figure 29 shows the velocity prediction for the 10^{th} second using Group G signals and plotted against the target value.

Table 9:Singals used in group G

Data Group G	Signals
	Current Velocity
	GPS
	Previous 5 Seconds
	EGO and Engine Parameters
	Segment Speed

Signals
Group B
Segment Speed

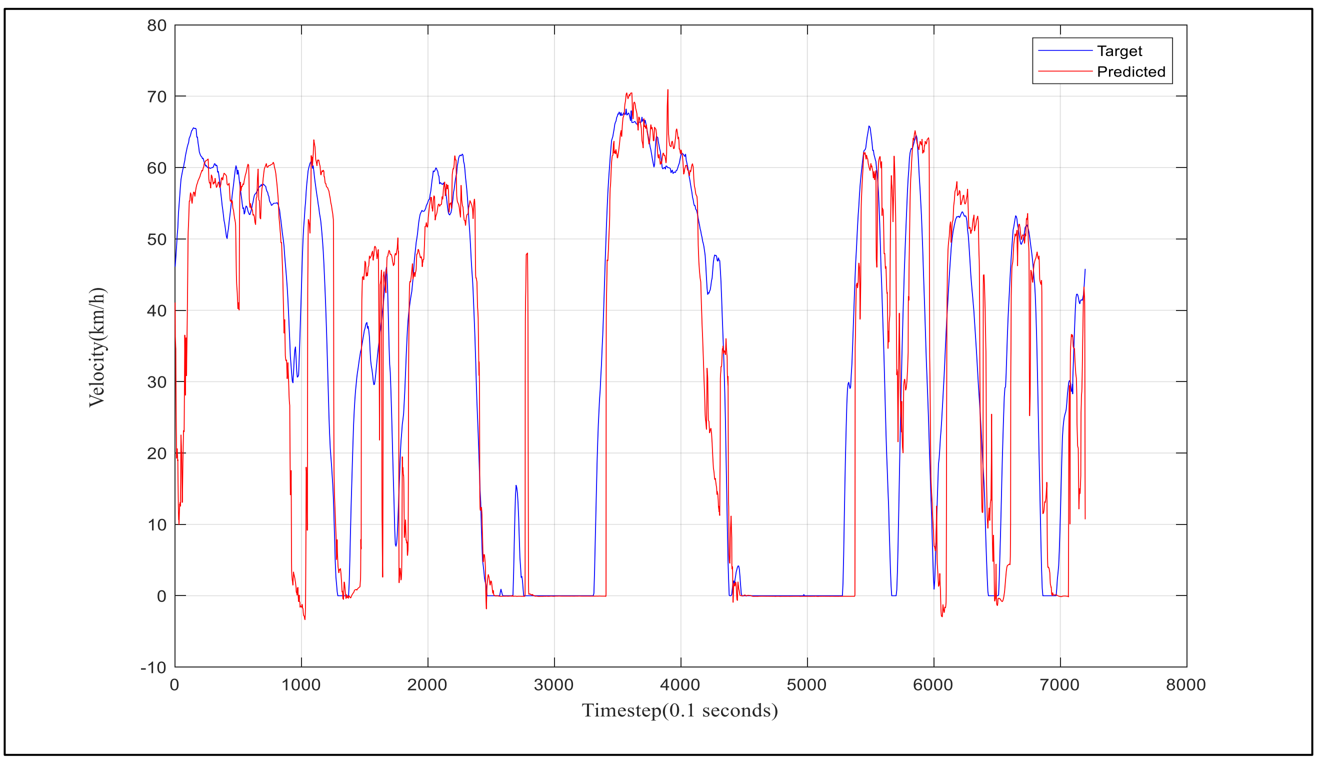


Figure 29: 10th-second prediction results using group G signals

Cross-validation results for the effect of signals on the prediction window:

Box and whiskers plot for MAE is shown in Figure 30 for every model for cross-validation results. Table 10 shows the result values. The horizontal axis represents different models, and the vertical axis represents MAE.

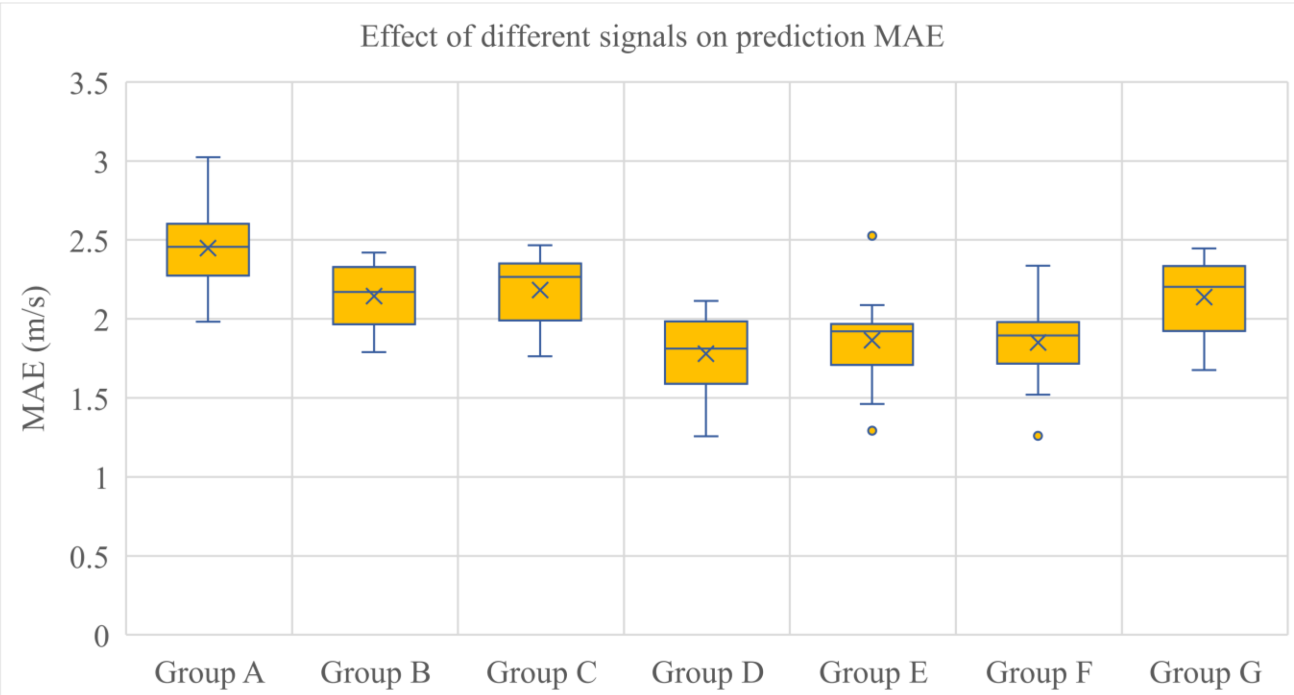


Figure 30: MAE results for different ***signals*** *for 10th-second prediction*

Table 10: MAE results for different signals for 10th -second prediction

	Group A	**Group B**	**Group C**	**Group D**	**Group E**	**Group F**	**Group G**
Test1	2.612	2.208	2.274	1.692	1.976	1.864	2.445
Test2	2.300	2.028	2.044	1.811	2.525	2.337	2.148
Test3	2.593	2.395	2.393	1.957	1.769	1.895	2.210
Test4	2.455	2.420	2.422	1.913	1.853	2.030	2.331
Test5	2.559	2.246	2.307	1.933	1.890	1.928	2.204
Test6	2.475	2.171	2.190	2.011	1.922	1.840	2.278
Test7	1.983	1.859	1.849	1.485	1.922	1.869	1.769
Test8	3.024	2.384	2.465	2.113	1.959	1.910	2.337
Test9	2.249	1.903	1.938	1.329	1.462	1.521	1.849
Test10	2.172	1.789	1.763	1.256	1.291	1.260	1.677
Test11	2.366	2.096	2.173	1.789	1.649	1.592	1.999
Test12	2.360	2.093	2.303	2.055	2.088	1.915	2.119
Test13	2.675	2.272	2.265	1.796	1.928	2.101	2.443
Average	2.448	2.143	2.184	1.780	1.864	1.851	2.139

Figure 31 represents the box and whiskers plot, and Table 11 result values for time shift for every model for cross-validation results. It can be observed that groups with SPaT data perform better in terms of time shift.

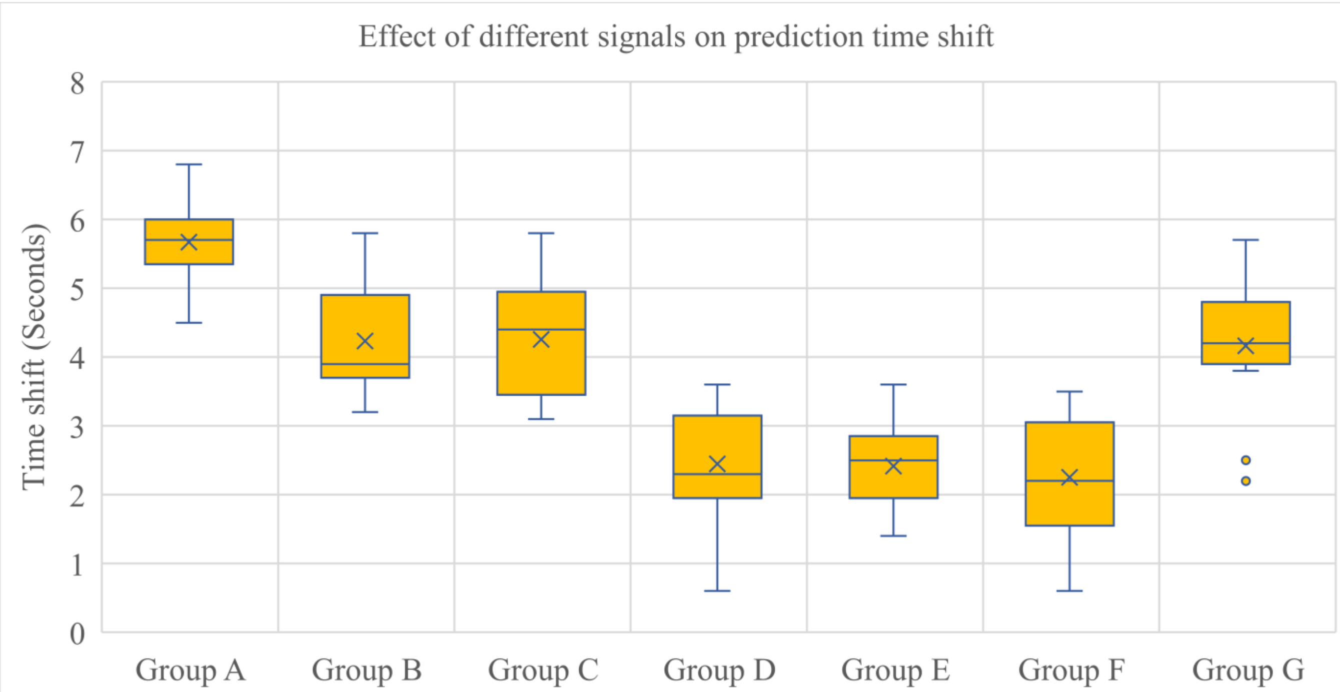


Figure 31: Time shift results for different ***signals*** *for 10th-second prediction*

Table 11: Time shift results for different signals for 10th -second prediction

	Group A	Group B	Group C	Group D	Group E	Group F	Group G
Test1	55	39	41	19	26	22	43
Test2	52	37	33	20	20	14	38
Test3	59	46	45	31	31	29	42
Test4	60	50	49	21	15	18	47
Test5	66	58	58	36	26	32	57
Test6	68	53	50	32	25	32	51
Test7	45	35	31	24	25	14	22
Test8	60	48	51	34	36	35	49
Test9	55	38	45	30	19	23	41
Test10	55	37	36	23	24	22	41
Test11	46	32	31	6	14	6	25
Test12	57	40	39	23	32	29	40
Test13	59	37	44	19	21	17	45
Average	56.69	42.31	42.54	24.46	24.15	22.54	41.62

3.2.1 Discussion on the Effect of Different Signals on Prediction

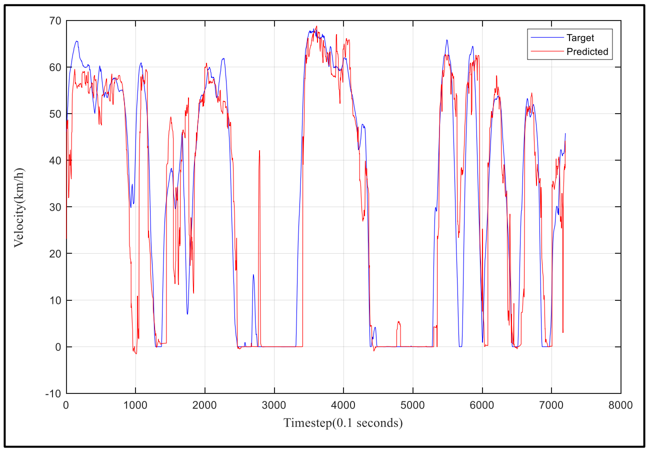
Signals are divided into different groups to assess the effect of different signals on prediction. It was observed in the literature that LSTM had performed better compared to other prediction models. Research conducted at the University of Michigan explored a variety of perception models. They studied auto-regressive moving average, shallow NN, long short-term memory (LSTM) deep NN, Markov chain, and conditional linear Gaussian models for prediction accuracy. Their study concluded that the LSTM deep NN provided the best prediction fidelity (measured in mean absolute error) [31]. Hence, these signals are used as input to the LSTM model for getting velocity prediction.

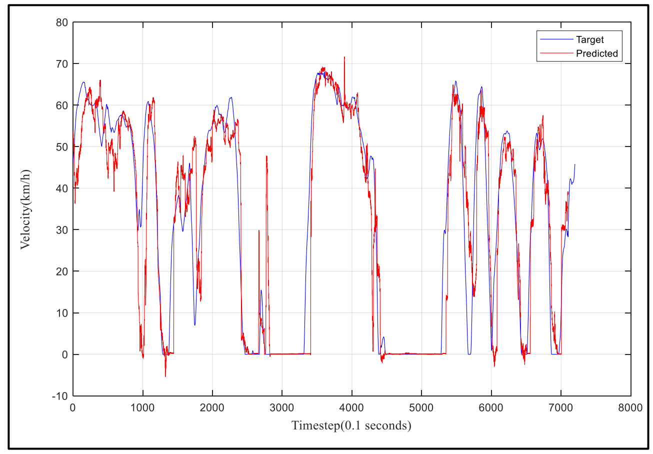
In cross-validation, prediction for each drive instance is obtained, such that the remaining 12 are used for prediction. Using these results, box and whiskers plot is obtained for both MAE and Timeshift. The box and whiskers plot divide the result into four quartiles. The two whiskers represent the extreme ends of results. The line within the box represents the median of the results, separating two quartiles on both sides, i.e., 50% of results lye below the median line. The cross represents the mean value of the results.

It is important to understand that GPS and current velocity are most important for velocity prediction. It is one of the essential parameters to understand the velocity profile and understand the location of the vehicle. Hence it has been kept common in all the groups. We can observe that, in both cases, the prediction results for group D, group E, and group F shows the lowest average values for MAE and time shift. Among these groups, group D shows the lowest MAE. One common signal used in this group is SPaT data. These results, therefore, show that the SPaT data is beneficial for getting an accurate prediction. On the other hand, segment speed data may worsen the prediction results, as observed in group E results. Segment speed is the average velocity of vehicles traveling at a certain point on the map. This point may not be relevant to the current position of the vehicle. Therefore, usage of Segment speed data may result in poor prediction.

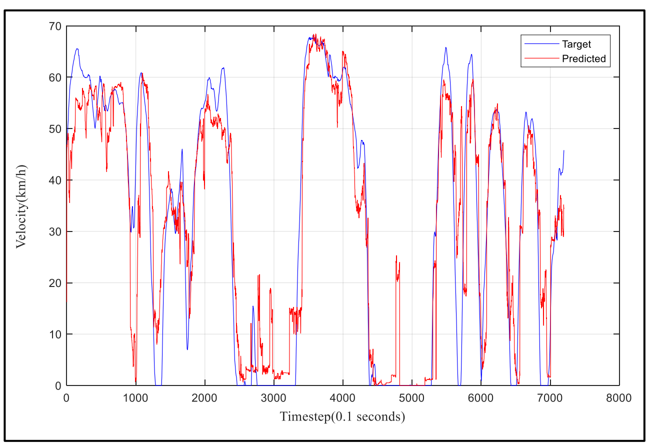
3.3 Effect of Different Models

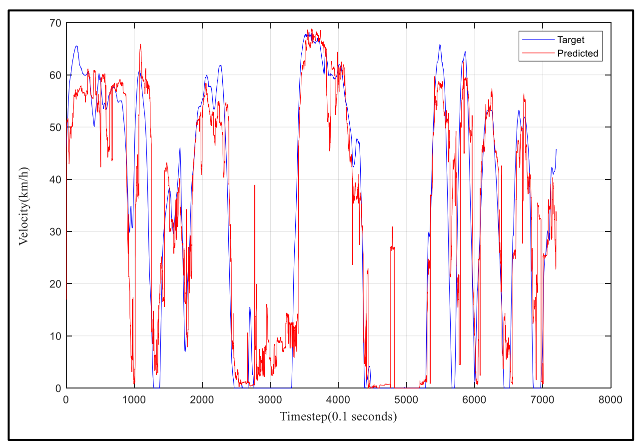
We can understand from 3.2 that Group D signals perform better than other groups. These group D signals are used as input to the different models for getting the 10th-second prediction. We can observe in Figure 32 that LSTM and CNN perform better, while Extra trees and Random forest shows some misprediction.





(a) (b)





(c) (d)

Figure 32: 10th -second prediction for (a) LSTM (b) CNN (c) Extra Trees (d) Random Forest

Box and whiskers plot for MAE is shown in Figure 33 for every model for cross-validation results. Table 12 shows the result values. The horizontal axis represents different models, and the vertical axis represents MAE. The significance of the line within the box is that two-quarters of the values lie below the line. The whiskers represent upper and lower values found during cross-validation.

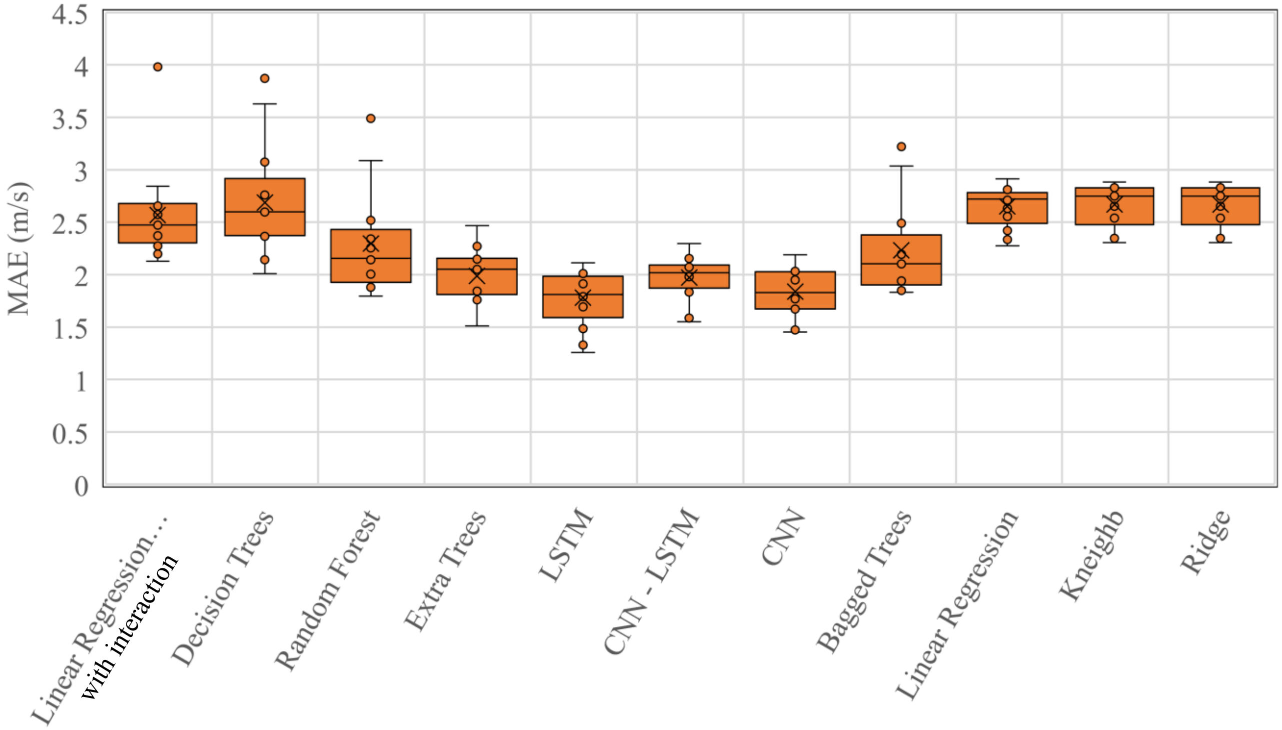


*Figure 33: MAE results for different **models** for 10th-second prediction*

Table 12: MAE results for different models for 10th -second prediction

	LR with interaction	**Decision Trees**	**Random Forest**	**Extra Trees**	**LSTM**	**CNN LSTM**	**CNN**	**Bagged Trees**	**Linear Regression**	**KNN**	**Ridge**
Test 1	2.84	2.60	2.34	2.15	1.69	2.09	1.99	2.27	2.81	2.83	2.83
Test 2	2.20	2.38	2.25	2.05	1.81	2.00	1.80	2.19	2.75	2.80	2.80
Test 3	2.40	3.87	3.49	2.47	1.96	2.02	1.83	3.22	2.91	2.88	2.88
Test 4	2.47	2.41	2.00	1.81	1.91	2.10	1.95	1.90	2.72	2.75	2.75
Test 5	2.47	3.63	3.09	2.12	1.93	1.98	2.19	3.04	2.76	2.83	2.83
Test 6	2.70	2.43	2.14	2.06	2.01	1.91	2.02	2.10	2.55	2.54	2.54
Test 7	2.27	2.36	1.88	1.76	1.49	1.83	1.67	1.85	2.33	2.35	2.35
Test 8	2.33	2.76	2.52	2.27	2.11	2.30	2.03	2.49	2.75	2.71	2.71
Test 9	2.13	2.63	2.16	1.81	1.33	1.58	1.47	2.01	2.42	2.41	2.41
Test 10	3.98	2.01	1.79	1.51	1.26	1.55	1.45	1.83	2.28	2.30	2.30
Test 11	2.37	2.14	1.91	1.82	1.79	2.15	1.67	1.94	2.83	2.85	2.85
Test 12	2.66	3.07	2.34	2.17	2.06	2.08	2.05	2.26	2.71	2.77	2.77
Test 13	2.57	2.62	1.94	1.84	1.80	2.07	1.77	1.91	2.63	2.65	2.65
Average	2.57	2.69	2.30	1.99	1.78	1.97	1.84	2.23	2.65	2.67	2.67

Figure 34 represents the box and whiskers plot, and Table 13 result values for time shift for every model for cross-validation results. It can be observed that Machine learning models perform better in terms of time shift.

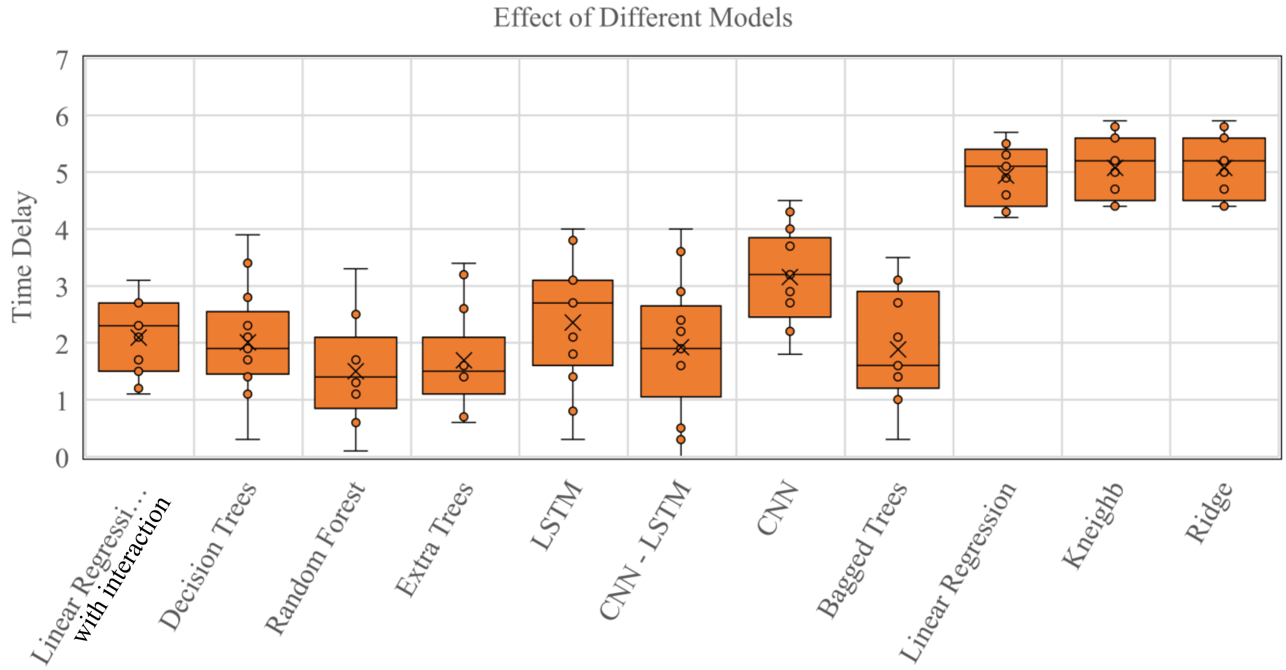


*Figure 34: Time Shift results for different **models** for 10th-second prediction*

Table 13: Time Shift results for different model for 10th-second prediction

	LR with Interacti	**Decision Trees**	**Random Forest**	**Extra Trees**	**LSTM**	**CNN - LSTM**	**CNN**	**Bagged Trees**	**Linear Regressi**	**KNN**	**Ridge**
Test 1	2.1	2.3	2.5	2.6	2.8	2.9	2.9	3.1	5.5	5.6	5.6
Test 2	2.3	1.8	1.2	1.6	2.8	2.3	3.3	1.6	5.1	5.2	5.2
Test 3	3.1	3.4	2.6	3.4	4	4	4.3	3.1	5.1	5.2	5.2
Test 4	1.5	1.7	1.4	1.6	2.1	0.3	2.9	1.7	4.9	5	5
Test 5	1.5	2.8	1.7	1.5	3.1	2.4	4.5	2.7	5.7	5.9	5.9
Test 6	2.3	1.9	1.3	1.5	3.1	2.2	3.3	1.4	5.3	5.6	5.6
Test 7	1.7	0.3	0.1	0.7	0.8	0.5	2.2	0.3	4.3	4.4	4.4
Test 8	2.3	3.9	3.3	3.2	3.8	3.6	3.7	3.5	5.6	5.8	5.8
Test 9	2.7	1.1	1.1	1.4	2.7	1.6	3.2	1.4	5.1	5.2	5.2
Test 10	2.7	1.5	0.6	0.6	1.4	1.9	2.2	1	4.2	4.4	4.4
Test 11	1.1	1.4	0.6	0.8	0.3	0	1.8	1	4.4	4.5	4.5
Test 12	2.7	2.1	1.4	1.5	1.9	1.6	4	1.6	4.6	4.7	4.7
Test 13	1.2	1.9	1.7	1.6	1.8	1.7	2.7	2.1	4.4	4.5	4.5
Average	2.09	2.01	1.50	1.69	2.35	1.92	3.15	1.88	4.94	5.08	5.08

3.3.1 Effect of Different Models on Prediction

From Section 3.2.1, we can understand that group D signals perform better, which contains the SPaT data. Now group D signals are used as input to different models to get the prediction. Similar to the previous case, results for each drive instance is obtained to get cross-validation results. These results are then plotted in box and whiskers plot for MAE and Timeshift.

In the case of MAE, deep neural networks perform better than machine learning models. Among the deep learning, model LSTM performs best in terms of MAE. CNN model performance is very close to LSTM. CNN-LSTM performs third best, but in the case of neural networks, it is very difficult to provide reasoning for why it did not perform the best. In the case of machine learning models, the extra trees model performed better, followed by the random forest model.

In the case of time shift, Machine learning models perform better than deep neural network models. Extra trees perform best in terms of time shift. The current velocity is an essential input to get an accurate velocity profile. It seems that machine learning models do not use velocity input on priority. For example, in the case of extra trees, feature performance was observed, and it shows that brake pedal is more important than current velocity. We can observe in Figure 32 that the machine learning models show more predict errors compared to DNN models. Hence more emphasis is given to the lower MAE than the lower time shift. Therefore, LSTM performed better compared to other models.

3.4. Effect on Prediction Window

3.4.1. Effect of Signal Groups on Prediction Window

Signal Groups are used to obtain the prediction results for the next 10, 15, 20, and 30 seconds of forward prediction. These results are assessed based on MAE, and the plot is shown in Figure 35. Table 14 represents the average result values for MAE. We can observe that as the target horizon increases, the MAE starts increasing. Group D results perform the best, as SPat data is beneficial for prediction. It can also observe that Group D shows almost 50% error reduction compared to Group A. Although Group E and Group F have more signals, the Segment speed did not improve the results. These are consistent with the results obtained in Asher, Zachary D [41], where less fuel economy was observed due to mispredictions when the Segment Speed signal is used for prediction.

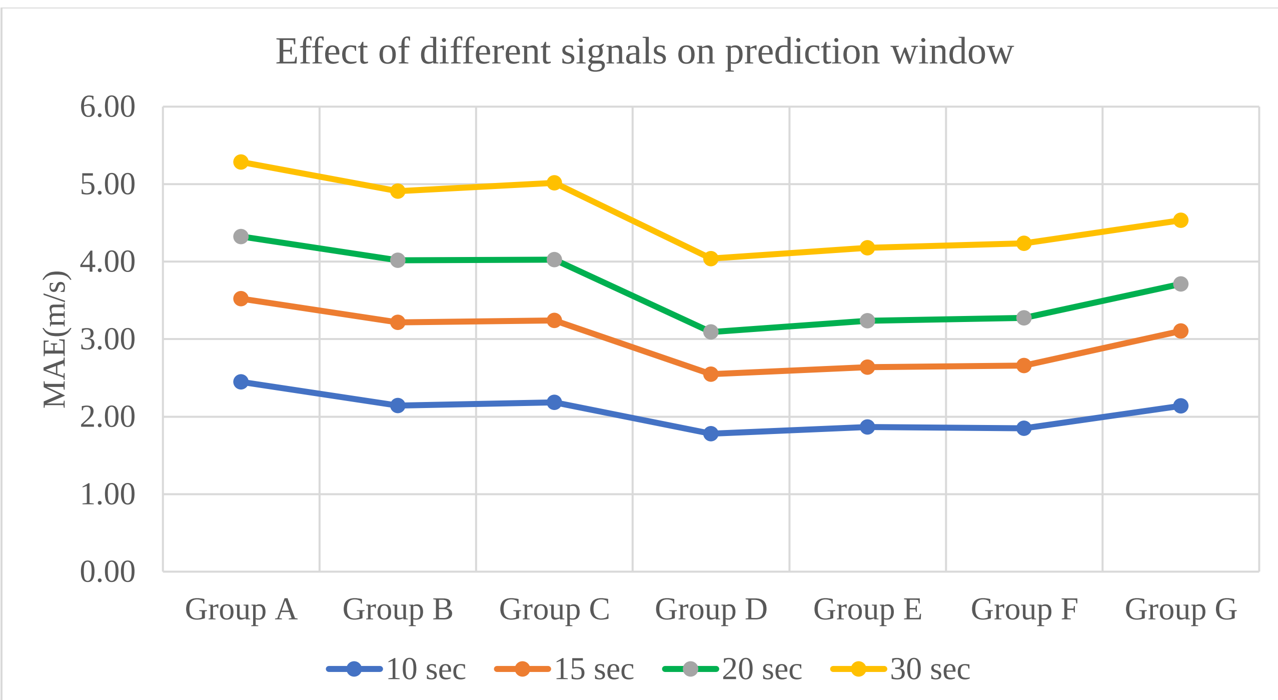


Figure 35: Effect of signal groups on MAE

Table 14: Effect of signal groups on MAE

Prediction Window	Group A	Group B	Group C	Group D	Group E	Group F	Group G
10 sec	2.45	2.14	2.18	1.78	1.86	1.85	2.14
15 sec	3.52	3.22	3.24	2.55	2.64	2.66	3.11
20 sec	4.32	4.02	4.02	3.09	3.24	3.28	3.71
30 sec	5.29	4.91	5.02	4.04	4.18	4.23	4.53

The effect due to signal groups on time shift is shown in Figure 36. The average result values are shown in Table 15. We can observe that, as the prediction window increases, the time shift in the prediction increases.

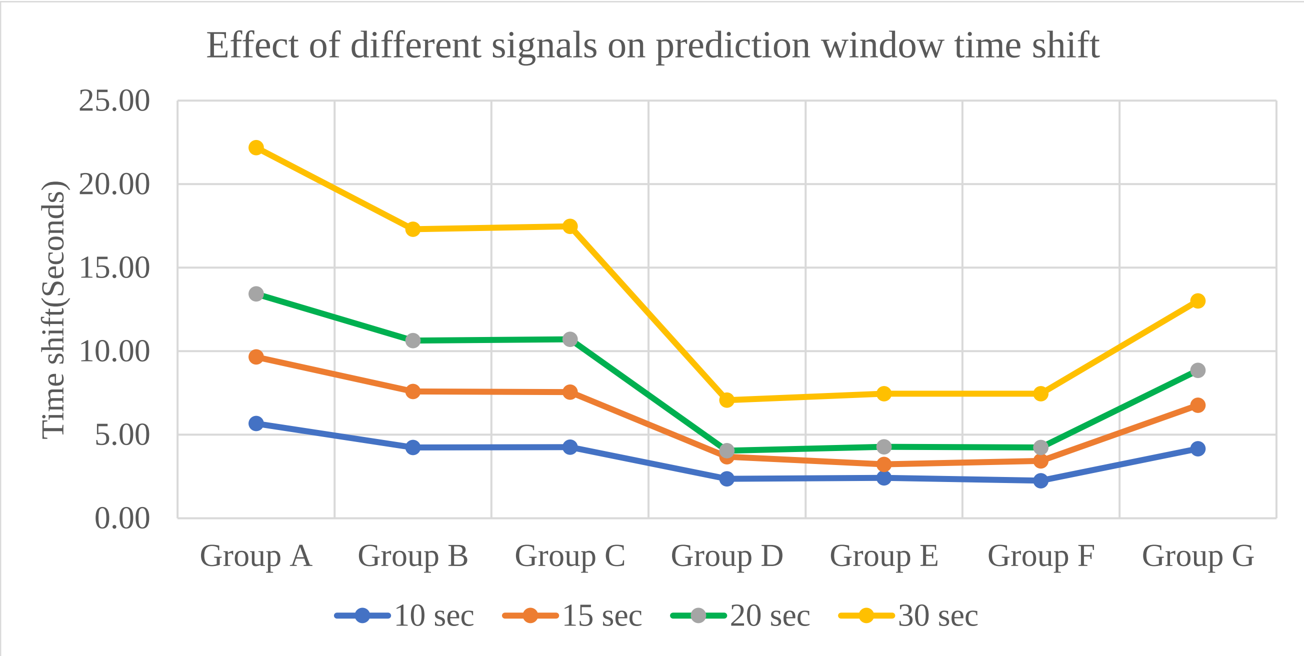


Figure 36: Effect of groups on time shift

Table 15: Effect of groups on time shift

Prediction Window	Group A	Group B	Group C	Group D	Group E	Group F	Group G
10 sec	5.67	4.23	4.25	2.35	2.42	2.25	4.16
15 sec	9.65	7.59	7.55	3.68	3.23	3.44	6.77
20 sec	13.43	10.64	10.72	4.05	4.27	4.23	8.85
30 sec	22.19	17.31	17.48	7.07	7.45	7.46	13.01

3.4.2. Effect of Different Models on the Prediction Window:

Different models are used for predicting 10, 15, 20, 30 seconds predictions. The effect of different models on MAE is shown in Figure 37. We can observe that as the prediction window increases, there is an increase in MAE. On the other hand, Machine learning models are consistent in terms of time shift, while deep neural networks still show an increase in time shift. The values for MAE and time shift are shown in Table 16 and Table 17.

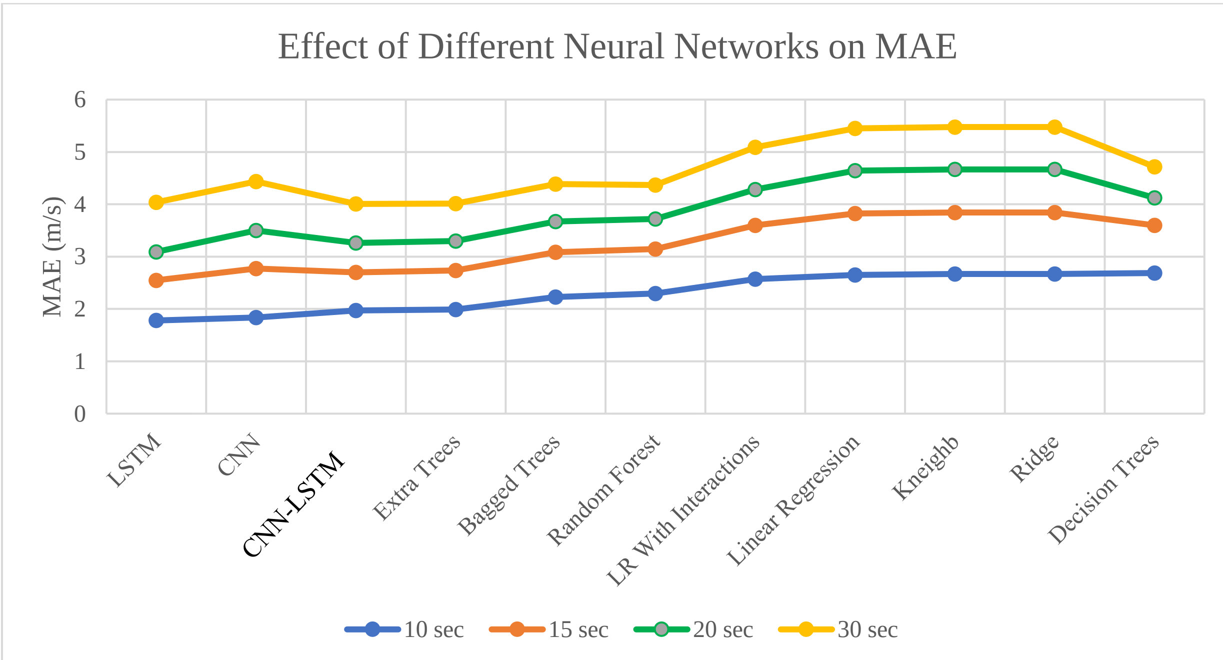


Figure 37: Effect of different neural networks on MAE

Table 16: Effect of different neural networks on MAE

Models	10 sec	15 sec	20 sec	30 sec
LSTM	1.78	2.55	3.09	4.04
CNN	1.84	2.77	3.50	4.43
CNN LSTM	1.97	2.70	3.26	4.01
Extra trees	1.99	2.73	3.30	4.02
Bagged trees	2.23	3.09	3.67	4.38
Random forest	2.30	3.15	3.72	4.37
LR with interactions	2.57	3.60	4.28	5.09
Linear regression	2.65	3.82	4.65	5.45
KNN	2.67	3.84	4.67	5.48
Ridge	2.67	3.84	4.67	5.48
Decision trees	2.69	3.60	4.12	4.72

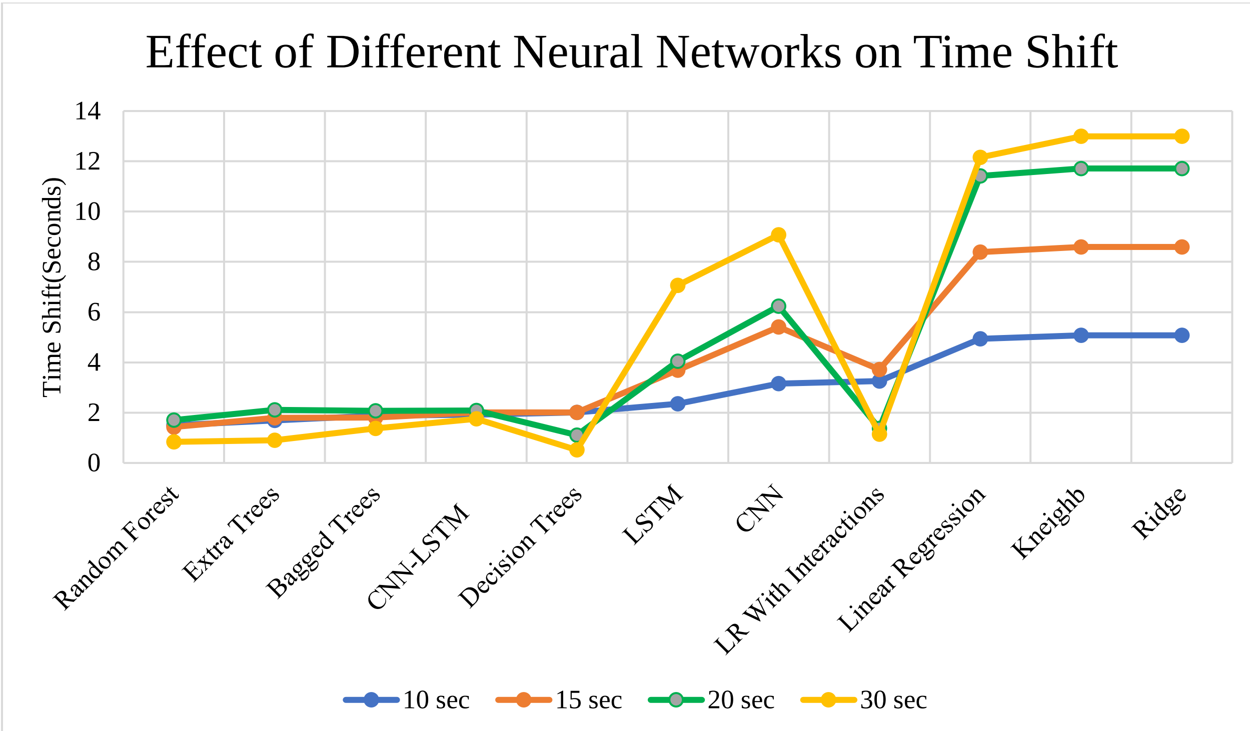


Figure 38:Effect of the different neural network on time shift

Table 17: Effect of different neural networks on time shift

MODEL	10 sec	15 sec	20 sec	30 sec
Random forest	1.50	1.43	1.70	0.84
Extra trees	1.69	1.79	2.11	0.90
Bagged trees	1.88	1.81	2.08	1.37
CNN -LSTM	1.92	2.02	2.08	1.75
Decision trees	2.01	2.01	1.11	0.52
LSTM	2.35	3.68	4.05	7.07
CNN	3.15	5.41	6.23	9.07
LR with interactions	3.26	3.72	1.37	1.14
Linear regression	4.94	8.39	11.42	12.16
KNN	5.08	8.59	11.72	12.99
Ridge	5.08	8.59	11.72	12.99

3.4.3 Discussion on the Effects on Prediction Window

To understand the effects of signals and prediction models, average prediction for 10, 15, 20 and 30 seconds are obtained. In the case of effects of signals, we can understand clearly that with the increase in prediction window, there is increase in MAE and time shift. Also group D, E and F containing SPaT signals show lowest MAE and time shift.

In case of effects of different model of prediction, there is increase in MAE with increase in prediction window. On the other hand, in case of time shift, the ensembled models show consistency with lower time shift. The time shift of 30 second prediction is even lower than 10, 15, and 20 seconds. DNN models show increase in time shift with increase in MAE. Models such as Linear regression, KNN, and Ridge regression is worse compared to other models.

3.5. 1- 10 seconds Forward Prediction.

Nowadays, many researchers are exploring the optimal EMS, which can be implemented practically, which is the motivation for this project, as discussed in section 1.3. Model predictive control (MPC) is a prospective strategy that can deliver optimal controls with practical considerations [42-44]. Model predictive control (MPC) involves control of a dynamic system based on different control inputs which are applied from the current time to a future time resulting in minimization of a cost function subject to the system dynamics and additional constraints [45-47]

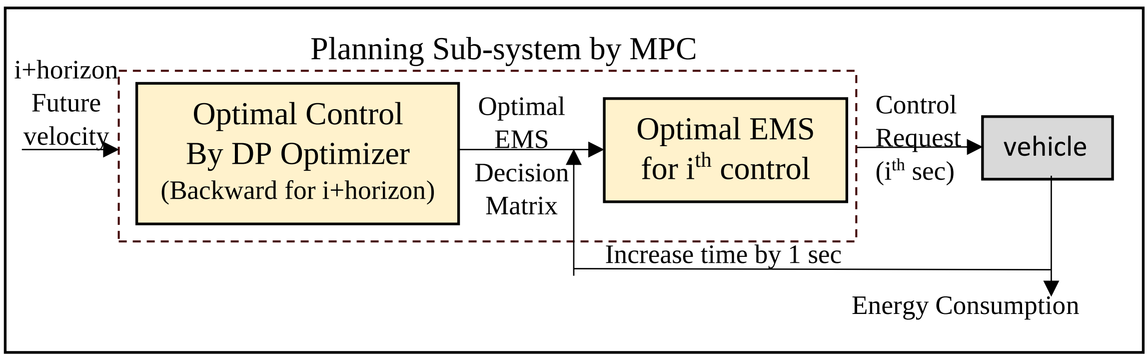
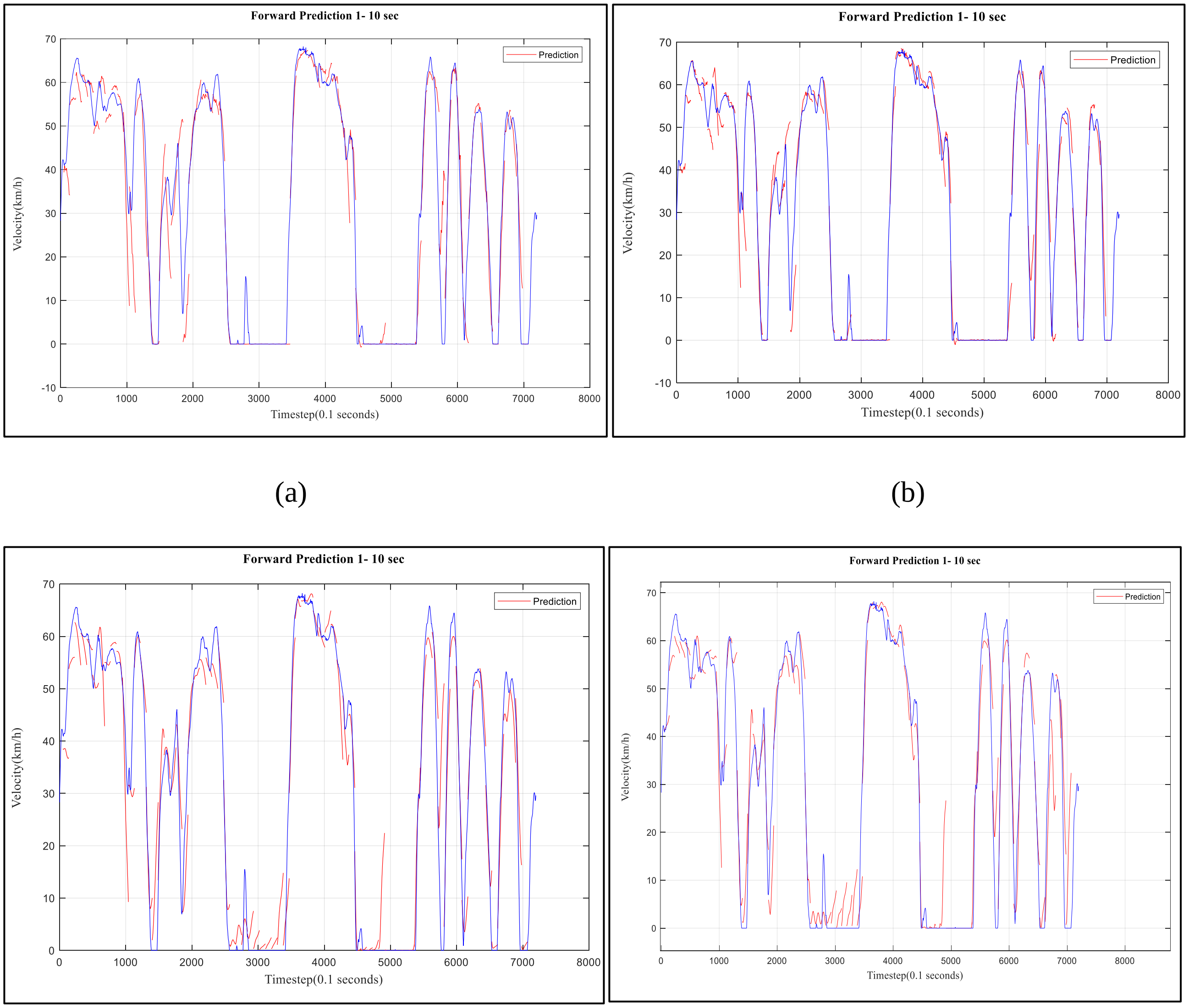


Figure 39: Detailed view of model predictive control

The detailed MPC planning sub-system is shown in Figure 39. It is difficult to obtain a prediction for the whole drive cycle in advance. But, we can get the velocity predictions for the desired horizon at each time index as we move forward. So, MPC takes the predicted velocity of the desired horizon as an input to calculate optimal controls only for that selected horizon using dynamic programming optimizer in backward in time. This process provides us optimal control for the entire horizon, which can result in fuel economy.

The forward prediction for every second from 1st second to 10th second is obtained from different neural networks. These forward prediction from 1- 10 seconds is plotted at every 10-second interval. So, at every 10-second interval, a hairy plot can be obtained, which is shown in Figure 40. In a hairy plot, we observe more distinct hairs, which are mispredictions. We can observe that there are few mispredictions for LSTM and CNN. While we see many mispredictions in Extra trees and Random Forest.



(a) (b)

(c) (b)

Figure 40: Velocity prediction 1-10 second ahead using (a) LSTM (b) CNN (c) Extra Trees (d) Random Forest

4. Summary

Overall following are the summary points of each study conducted in the research

- Effect of **Different Signals** on prediction.

  When grouped signals are given input to the LSTM neural network to predict 10-second prediction, it was observed that Group D signals perform better. It was observed that Group B signals perform better than Group A due to adding of the previous 5 seconds and EGO and Engine parameters. Group D data contains current velocity, GPS, previous 5 seconds of velocity, EGO, and Engine parameters, and Spat (Signal Phase and Timing). In the case of Group E and Group F, Segment speed tends to worsen prediction. Hence group E and group F show some mispredictions when compared with the group D performance. Overall, we observed that Group D shows an almost 50% error reduction compared to Group A. Group D as an input to LSTM used for 10^{th}-second prediction MAE of 1.78 m/s and the time shift of 2.35 seconds is observed.

- Effect of **Different Models** on prediction.

  Results are assessed based on MAE and time shift. Group D signals are used as input to various models to prediction 10^{th}-second velocity. We can observe that a deep neural network model performs better in terms of MAE, while classical machine learning models perform better in terms of time shift. Overall, we observed that the LSTM model performs best for vehicle velocity prediction since it can predict more accurately compared to other models. MAE of 1.78 is observed with the LSTM model for 10-second prediction. Random forest time shift 1.50 seconds for 10^{th}- seconds.

- Effect on **prediction window** by different signals and different models.

  When we predict velocity using different signals or different models, we observed an increase in MAE in both cases. Time shift increases for prediction using different signals. On the other hand, the time shift does not increase much in the case of ensembles. Overall 10^{th}-second prediction gives better accuracy.

- **1-10 seconds** forward prediction.

  Forward prediction of every second is obtained for 1 to 10 seconds and plotted on hairy plots for every 10-second interval. Overall it was observed that the LSTM and CNN model performed better compared to other machine learning models.

5. Conclusion

In this study, multiple signals collected in Fort Collins, Colorado, which are used in different Groups for getting a prediction for 10, 15, 20,30 seconds from LSTM. Group D dataset showed the best performance, where SPaT data was observed very useful for prediction. The assessment was done based on MAE and time shift. We also compared different models, which include deep neural networks and classical machine learning models. LSTM shows excellent performance for predicting vehicle velocity and can be used for predicting different prediction horizons. This study demonstrates that V2I data is essential to obtain significant improvement in vehicle velocity prediction. ANN's ability to predict future velocity is observed. It also shows the performance of different signals and models on the prediction.

In future work, this study could be extended with a higher number of drive cycles. We can also include V2V and Camera data and more V2I data in the dataset and study the combination in different groups. Prediction with respect to distance can be used by gathering data with respect to distance for predicting velocity in the forward distance. It also seeks to work on implementation where signals collected on NVIDIA Drive PX2 can be used for getting predictions.

References

1. Fagnant, Daniel J., and Kara Kockelman. "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations." Transportation Research Part A: Policy and Practice 77 (2015): 167-181.Energy-Efficient Driving of Road Vehicles:.

2. Waschl, Harald, Ilya Kolmanovsky, and Frank Willems. Control Strategies for Advanced Driver Assistance Systems and Autonomous Driving Functions. Springer Vlg, 2019.Nkoro, A.B. and Vershinin, Y.A., "Current and Future Trends in Applications of Intellige.

3. Birky, A., Laughlin, M., Tartaglia, K., Price, R. et al.,"Electrification Beyond Light Duty: Class 2b-3 Commercial Vehicles," No. ORNL/TM-2017/744. Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2018.

4. IDC's Worldwide Semiannual Internet of Things Spending Guide, May 2016 and Vehicles, connected cars are a subset of Connected.

5. Automated Driving, Levels of Driving automation defined in new SAE International Stanadard J3016.

6. Seeker. "How Close Are We to a Self-Driving World?". Filmed [Apr 12, 2019]. YouTube video Posted [Apr 12, 2019]. https://youtu.be/U5laBg-ERbQ.

7. EIA, 2017: May 2017 Monthly Energy Review. DOE/EIA-0035(2017/5). U.S. Department of Energy, U.S. Energy Information Administration (EIA), Washington, DC, 243 pp.

8. Reidmiller, D. R., C. W. Avery, D. R. Easterling, K. E. Kunkel, K. L. M. Lewis, T. K. Maycock, and B. C. Stewart. "Impacts, Risks, and Adaptation in the United States: Fourth National Climate Assessment, Volume II." (2017).

9. Pierre-Louis, Kendra " Greenhouse Gas Emissions Accelerate Like a 'Speeding Freight Train' in 2018" The new York times [New York] Dec. 6, 2018 Published: Page A1.

10. UNFCCC, 2015: Paris Agreement. United Nations Framework Convention on Climate Change, [Bonn, Germany], 25 pp.

11. Atabani, A.E., Badruddin, I.A., Mekhilef, S., Silitonga, A.S.: A review on global fuel economy standards, labels and technologies in the transportation sector. Renewable Sustainable Energy Rev. 15, 4586–4610 (2011).

12. Bender, F.A., Kaszynski, M., and Sawodny, O., "Drive Cycle Prediction and Energy Management Optimization for Hybrid Hydraulic Vehicles," IEEE Trans Veh Technol 62:3581-3592, 2013.

13. Phillips, Derek J., Real-time Prediction of Automotive Collision Risk from Monocular Video (2019)).

14. Gaikwad, Tushar D., Zachary D. Asher, Kuan Liu, Mike Huang, and Ilya Kolmanovsky. Vehicle Velocity Prediction and Energy Management Strategy Part 2: Integration of Machine Learning Vehicle Velocity Prediction with Optimal Energy Management to Improve.

15. Bernard Marr. “What is the Difference Between Artificial Intelligence and Machine Learning?” Time, Dec 6, 2016. https://www.forbes.com/sites/bernardmarr/2016/12/06/what-is-the-difference-between-artificial-intelligence-and-machine-learning/.

16. What is Deep Learning by Jason Brownlee, August 2019 in Deep Learning, Machine learning mastery.

17. Gong Q, Li Y, Peng ZR (2008) Trip-Based Optimal Power Management of Plug-in Hybrid Electric Vehicles. IEEE Trans Veh Technol 57:3393–3401.

18. Gong Q, Li Y, Peng Z (2009) Power management of plug-in hybrid electric vehicles using neural network based trip modeling. In: 2009 American Control Conference. ieeexplore.ieee.org, pp 4601–4606.

19. Mohd Zulkefli MA, Zheng J, Sun Z, Liu HX (2014/8) Hybrid powertrain optimization with trajectory prediction based on inter-vehicle-communication and vehicle-infrastructure-integration. Transp Res Part C: Emerg Technol 45:41–63.

20. Lefèvre, S., Sun, C., Bajcsy, R., and Laugier, C., “Comparison of Parametric and Non-Parametric Approaches for Vehicle Speed Prediction,” in American Control Conference, June 4-6, 2014, doi:10.1109/ACC.2014.6858871.

21. Sun C, Hu X, Moura SJ, Sun F (2015) Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles. IEEE Trans Control Syst Technol 23:1197–1204.

22. Sun C, Moura SJ, Hu X, Hedrick JK, Sun F (2015) Dynamic Traffic Feedback Data Enabled Energy Management in Plug-in Hybrid Electric Vehicles. IEEE Trans Control Syst Technol 23:1075–1086.

23. Lemieux, J. and Ma, Y., “Vehicle Speed Prediction Using Deep Learning,” in IEEE Vehicle Power and Propulsion Conference, 1-5, 2015.

24. Rezaei, Amir, and Jeffrey B. Burl. "Prediction of vehicle velocity for model predictive control." IFAC-PapersOnLine 48, no. 15 (2015): 257-262.

25. Hellstrom, E., and Jankovic, M., “A Driver Model for Velocity Tracking with Look-Ahead”, in American Control Conference, Chicago, IL, 2015.

26. Baker D, Asher Z, Bradley T (2017) Investigation of Vehicle Speed Prediction from Neural Network Fit of Real World Driving Data for Improved Engine On/Off Control of the EcoCAR3 Hybrid Camaro. SAE Technical Paper.

27. Olabiyi, O., Martinson, E., Chintalapudi, V., Guo, R., “Driver Action Prediction Using Deep (Bidirectional) Recurrent Neural Network”, eprint arXiv:1706.02257, 2017.

28. Zhang, F., Xi, J., and Langari, R., “Real-Time Energy Management Strategy Based on Velocity Forecasts Using V2V and V2I Communications,” IEEE Transactions on Intelligent Transportation Systems 18:416-430, 2017.

29. Sun C, Sun F, He H (2017) Investigating adaptive-ECMS with velocity forecast ability for hybrid electric vehicles. Appl Energy 185, Part 2:1644–1653.

30. Baker, David, Zachary Asher, and Thomas Bradley. Investigation of vehicle speed prediction from neural network fit of real world driving data for improved engine On/Off control of the EcoCAR3 hybrid camaro. No. 2017-01-1262. SAE Technical Paper, 2017.

31. Liu K, Asher Z, Gong X, Huang M, Kolmanovsky I (2019) Vehicle Velocity Prediction and Energy Management Strategy Part 1: Deterministic and Stochastic Vehicle Velocity Prediction Using Machine Learning.

32. Jason Brownlee, Long Short-Term Memory Networks With Python (Machine learning mastery).

33. Olah, C. et al., "Understanding LSTM Networks," 2015, http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

34. LSTM and RNN Tutorial with Demo (with Stock/Bitcoin Time Series Prediction, Sentiment Analysis, Music Generation) https://github.com/omerbsezer/LSTM_RNN_Tutorials_with_Demo.

35. Huang, Chiou-Jye, and Ping-Huan Kuo. "A deep cnn-lstm model for particulate matter (PM2. 5) forecasting in smart cities." Sensors 18, no. 7 (2018): 2220.

36. Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." arXiv preprint arXiv:1803.01271 (2018).

37. Should We Abandon LSTM for CNN? by Geoffrey So, Mar 29 2019, medium.com.

38. Hadelin de Ponteves. "Deep Learning A-Z™: Hands-On Artificial Neural Networks" June 9, 2019, https://www.udemy.com/course/deeplearning/.

39. Jason Brownlee, Machine learning Mastery with python (Machine learning mastery).

40. Geurts, Pierre, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees." Machine learning 63, no. 1 (2006): 3-42.

41. Asher, Zachary D., Jordan A. Tunnell, David A. Baker, Robert J. Fitzgerald, Farnoush Banaei-Kashani, Sudeep Pasricha, and Thomas H. Bradley. Enabling Prediction for Optimal Fuel Economy Vehicle Control. No. 2018-01-1015. SAE Technical Paper, 2018.

42. Xie, Shaobo, Xiaosong Hu, Zongke Xin, and James Brighton. "Pontryagin's minimum principle-based model predictive control of energy management for a plug-in hybrid electric bus." Applied energy 236 (2019): 893-905.

43. Borhan, Hoseinali, Ardalan Vahidi, Anthony M. Phillips, Ming L. Kuang, Ilya V. Kolmanovsky, and Stefano Di Cairano. "MPC-based energy management of a power-split hybrid electric vehicle." IEEE Transactions on Control Systems Technology 20, no. 3 (201.

44. Fu, Lina, Ö. Ümit, Pinak Tulpule, and Vincenzo Marano. "Real-time energy management and sensitivity study for hybrid electric vehicles." In Proceedings of the 2011 American Control Conference, pp. 2113-2118. IEEE, 2011.

45. Meyer, Richard, Raymond A. DeCarlo, Peter H. Meckl, Chris Doktorcik, and Steve Pekarek. "Hybrid model predictive power flow control of a fuel cell-battery vehicle." In Proceedings of the 2011 American Control Conference, pp. 2725-2731. IEEE, 2011.

46. Meyer, Richard T., Raymond A. DeCarlo, Peter H. Meckl, Chris Doktorcik, and Steve Pekarek. "Hybrid model predictive power management of a fuel cell-battery vehicle." Asian Journal of Control 15, no. 2 (2013): 363-379.

47. Meyer, Richard T., Raymond A. DeCarlo, and Steve Pekarek. "Hybrid Model Predictive Power Management of a Battery-Supercapacitor Electric Vehicle." Asian Journal of Control 18, no. 1 (2016): 150-165.