Vehicle Maneuver Prediction
using Deep Learning Networks

by

Song Wang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical and Computer Engineering)
in the University of Michigan – Dearborn

2023

Doctoral Committee:

Professor Yi L. Murphey, Chair
Associate Professor Paul Watta
Associate Professor Wencong Su
Assistant Professor Mohamed Abouelenien

DEDICATION

This edition of the Ph.D. final dissertation written report by Song Wang, “VEHICLE MANEUVER PREDICTION USING DEEP LEARNING NETWORKS”, is dedicated to all the professionals, students and people who are working on or interested in power electronic, power system reliability, renewable energy, and complex network information entropy theory

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor, Yi Lu Murphey, who has been an exceptional mentor and guide. Her insight and expertise have been invaluable, and I am grateful for her patience and guidance throughout this process. I also want to thank the other members of my dissertation committee, who have provided valuable feedback, suggestions, and guidance throughout this process.

I would like to acknowledge the support of my colleagues and friends, who have been with me through this journey and have provided me with encouragement, feedback, and motivation.

Finally, I would like to thank my wife Yi Zhang for always being there for me, providing strong support both for my life and my research; and my parents for giving birth to me, providing steadfast support and unconditional love ever since then.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF FIGURES vi

LIST OF TABLES viii

ABSTRACT ix

Chapter 1: Introduction to Vehicle Maneuver Prediction 1

1.1 Background and Objectives 1

1.2 Problem Formulation 8

1.3 Overview of the Dissertation Structure 9

Chapter 2: Learning Temporal Dependencies in Driving Maneuvering 10

2.1 Introduction 10

2.2 Related Work 12

2.2.1 Recurrent Neural Networks 12

2.2.2 Transformer Models 15

2.3 VMP Systems Using Deep Sequential Learning Algorithm 17

2.3.1 LSTM Network based VMP system 17

2.3.2 Transformer Network based VMP system 20

2.4 Experiment Setup 24

2.4.1 Datasets 24

2.4.2 Evaluation Metrics 30

2.4.3 Baseline Models 31

2.4.4 Hyper-parameter Settings in VMP-L and VMP-T 31

2.5 Experiment Results 32

2.6 Discussion and Conclusion 33

Chapter 3: Leveraging Visual Data for Predicting Vehicle Maneuvers 35

3.1 Introduction 35

3.2 Related Work 40

3.2.1 Attention Mechanisms Related to VMP systems 40

3.2.2 The Use of Driver’s View Videos in VMP System 43

3.3 Visual Feature Extraction for VMP systems 45

3.3.1 Attention Feature Extraction 45

3.3.2 Driver Face Shift Feature Extraction 49

3.4 Experiment Setup 51

3.4.1 Dataset and Evaluation metrics 51

3.4.2 Baseline models 52

3.4.3 Hyper-parameter Settings 53

3.5 Experiment Results 53

3.6 Discussion and Conclusion 55

Chapter 4: Target Vehicle Position Prediction using V2V Communication and Deep Learning Neural Networks 56

4.1 Introduction 56

4.2 Related Work 60

4.3 Extracting Effective Features For Remote Vehicle Prediction 62

4.4 Deep Learning Networks For Accurate Prediction Of Relative Positions Of Remote Vehicles 65

4.4.1 LSTM Network Models for Prediction of Relative Position of a remote vehicle 65

4.4.2 Attention focused transformer models for prediction of relative position of a remote vehicle . 66

4.5 Experiments Setup 67

4.5.1 Hyper-parameter settings 67

4.5.2 V2V data used in experiments 67

4.5.3 Three Evaluation Methods 71

4.6 Experiment results 73

4.7 Conclusion 75

Chapter 5: Conclusions and Future Directions 77

References 80

LIST OF FIGURES

Figure 1. Forward collision warning (FCW) system. 2

Figure 2. Interactions between target vehicle and other vehicles. 3

Figure 3. Vehicle maneuver prediction on in-vehicle sensors' data, V2V data, and V2I data...... 6

Figure 4. Illustration of vehicle maneuver prediction.. 8

Figure 5. Architecture of a multilayered LSTM network. 18

Figure 6. LSTM cell.. 19

Figure 7. VMP-T: an attention focused transformer network for vehicle maneuver prediction. . 25

Figure 8. Traffic flow of EV and 4 SVs in 5 seconds on I-80 highway. 26

Figure 9. Workflow of sequence data extraction for lane change maneuver............................. 27

Figure 10. Target vehicle maneuver prediction using V2I data in NGSIM dataset. 28

Figure 11. Statistics and examples of ISL-UMD dataset.. 29

Figure 12. Ego vehicle maneuver prediction using in-vehicle sensors' data in UMD-ISL.......... 30

Figure 13. System comparison among various prediction horizons. 34

Figure 14. A heat map generated from a driving video frame. 37

Figure 15. Convolutional operations on input matrix.. 45

Figure 16. Max pooling operation. 47

Figure 17. A left-turn maneuver. 50

Figure 18. Ego vehicle maneuver prediction using in-vehicle sensors' data in UMD-ISL.......... 52

Figure 19. System comparison among various prediction horizons on UMD-ISL dataset..........54

Figure 20. V2V Communicaiton... 57

Figure 21. Overview of the proposed intelligent system for remote vehicle. 62

Figure 22. Illustration of eight position classes. 63

Figure 23. Normalized position of both the host and remote vehicles. 64

Figure 24. An illustration of the positional information. 65

Figure 25. Architecture of a multilayered LSTM network with input features. 66

Figure 26. VMP-T 66

Figure 27. A histogram of the closest distances between the pairs of a host and a remote vehicle 69

Figure 28. Illustration of relative positions of a host and a remote vehicle. 69

Figure 29. A histogram of the amount of time that the host vehicle is to the remote vehicle 70

Figure 30. Distribution of data samples in each class. 70

Figure 31. Remote vehicle position prediction using V2V data. 71

LIST OF TABLES

Table 2.1 Vehicles and trip durations in each location in NGSIM dataset 27

Table 2.2: F1 score for vehicle maneuver prediction on NGSIM dataset 34

Table 2.3: F1 score for vehicle maneuver prediction on UMD-ISL dataset 35

Table 3.1: F1 score for vehicle maneuver prediction on UMD-ISL dataset 55

Table 4.1: Statistics of the test set containing 4 complete trips 73

Table 4.2: Numbers of trips for different predict time 74

Table 4.3: Overall performances of systems using 3F by averaging three methods 75

Table 4.4: Overall performances of systems using 9F by averaging three methods 75

Table 4.5: Overall performances of systems using 11F by averaging three methods 75

Table 4.6: Comparison Of Effectiveness Features Used In Rv Position Prediction At Various Prediction Horizons 76

ABSTRACT

Vehicle maneuver prediction plays an important role in ADAS (Advanced Driver Assistance Systems) and autonomous vehicles. It predicts the future behaviors of surrounding vehicles based on the current and past driving states of vehicles. Accurately predicting a vehicle's future trajectory and maneuver intentions is essential for safe and efficient navigation in traffic. Compared to conventional physics-based models, deep learning approaches are getting more popular due to their better performances in complicated real-world scenarios. This dissertation studies the temporal and spatial dependencies of vehicle maneuvers in a driving trip and investigate an innovative deep learning system to predict maneuvers of surrounding vehicles. Our method utilizes a combination of sensor data such as GPS, speed, acceleration, and videos to predict the future maneuver of a vehicle. The system contains LSTM (Long Short-Term Memory) or Transformer networks to learn information from past driving states, and graph neural networks to exploit the spatial relations between surrounding vehicles.

We evaluate the proposed method on a large-scale real-world dataset and compare its performance with several state-of-the-art approaches. Our results show that our method significantly outperforms existing methods in terms of accuracy and robustness. In addition to the prediction performance, we also analyze the interpretability of the proposed method and demonstrate how it can be used to identify critical factors affecting maneuver prediction.

This research provides a significant contribution to the field of vehicle maneuver prediction and lays the foundation for the development of advanced ADAS and autonomous driving systems.

Our method has the potential to improve the safety and efficiency of road transportation and can be used to support the deployment of autonomous vehicles in complex driving scenarios.

Chapter 1: Introduction to Vehicle Maneuver Prediction

1.1 Background and Objectives

Autonomous vehicles offer the promise of increased safety for human drivers, passengers, and pedestrians. An important technology in modern vehicles is Advanced Driver Assistance Systems (ADAS) that assists drivers in avoiding accidents. ADAS contributes not only to increased safety but also to improved traffic management. However, the number of traffic accidents still remains high. A highway traffic study [1] shows that over 19% of accidents in the U.S. in 2020 are due to inappropriate maneuvers, e.g., changing or merging lanes at the wrong time. In 2018, autonomous cars from Uber and Tesla involved in two traffic accidents also worried people about the safety of autonomous cars. But studies show that maneuvers can be detected, understood, and predicted based on vehicle dynamics and traffic environments [2] [3].

Vehicle maneuver prediction is an important research topic in intelligent vehicle systems. Accurate prediction of vehicle maneuvers can help ensure the safety and efficiency of road transportation, reduce the likelihood of accidents, and improve the overall driving experience. It provides in-vehicle prediction of potentially dangerous driving maneuvers, which enables drivers take necessary and timely actions to avoid accidents. Many important techniques have been developed for vehicle maneuvering detection and prediction, e.g. forward-collision warning (FCW) system in Fig.1. The Insurance Institute for Highway Safety (IIHS) has already observed a 27% reduction in front-to-rear crashes through this technology [4]. The development of effective vehicle maneuver prediction algorithms is therefore a key research area in the field of autonomous driving.

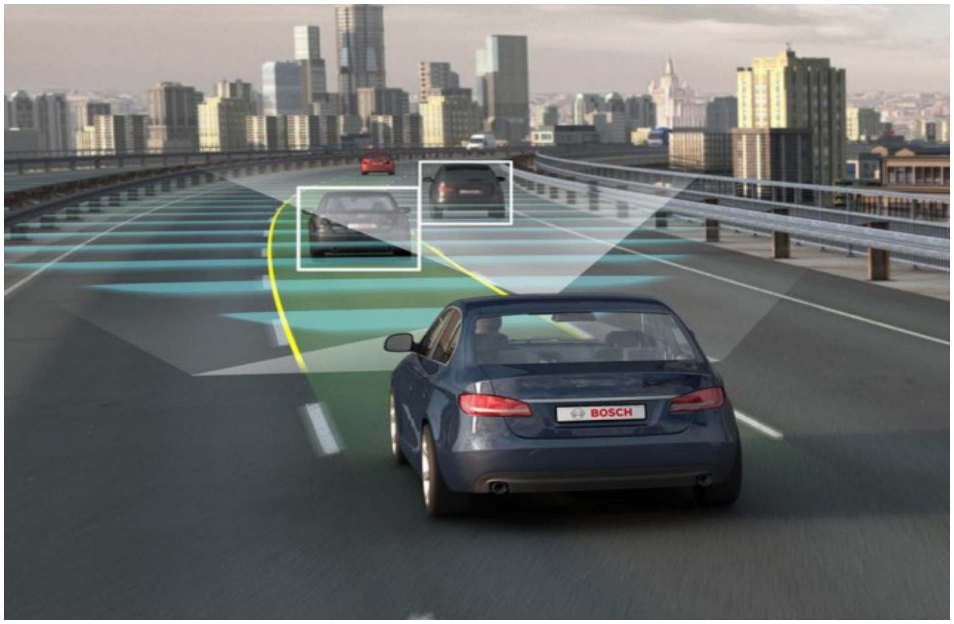


Figure 1. Forward collision warning (FCW) system alerts driver when the front car decelerate suddenly that may cause collision.

However it’s a challenging task since there are so many factors can affect vehicle’s maneuvers, such as traffic light, traffic signs, weather, vehicle dynamics, driver behavior, etc. Various technologies have been proposed to predict vehicle maneuvers, and mostly were developed using physics-based models or traditional machine learning technologies, such as Bayesian filtering methods [5], Support Vector Machine [6], and Hidden Markov Models [7]. However, these traditional methods were limited by their inability to capture the complex patterns and relationships in the data and were prone to errors in real-world scenarios.

With the advent of deep learning, researchers have explored the use of deep neural networks for vehicle maneuver prediction. Deep learning-based approaches leverage the powerful representation learning capabilities of neural networks to capture complex patterns and predict the future maneuver of a vehicle. Recent research in this field has proposed several deep learning-based approaches for vehicle maneuver prediction, including convolutional neural networks

(CNNs) [33, 34] and recurrent neural networks (RNNs) [35, 36]. Specifically, long short-term memory (LSTM) model and gated recurrent unit (GRU) are two popular variants of RNNs that are able to improve the accuracy in long-term prediction. They were introduced to prevent gradient vanishing or exploding problem in long sequence learning task and they are the most popular used RNNs now. LSTM and GRU have also been applied to predict vehicle maneuver in many work [8, 9, 10]. However, despite the advances in deep learning-based methods, there are still several challenges that need to be addressed in the field of vehicle maneuver prediction. One of the main challenges is the interpretability of the prediction models, which is crucial for ensuring the safety and reliability of autonomous vehicles. In addition, most of those work only consider the target vehicle they are going to predict. Their models take target vehicle's signals as input but don't consider other vehicles' signals. In real driving environment, the interactions among vehicles are important for predicting vehicle maneuvers. As shown in Fig.2, the target vehicle's maneuvering is limited by the positions and motions of surrounding vehicles, thus the states of other vehicles

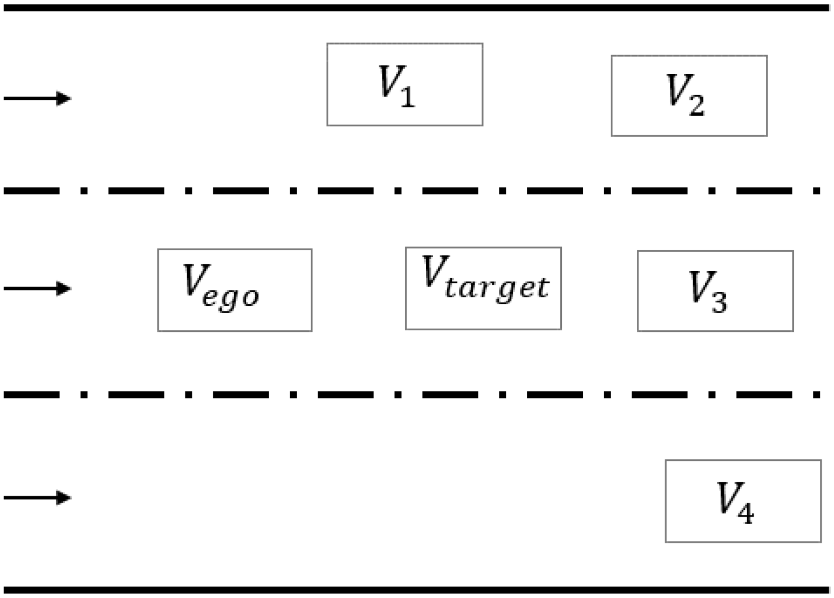


Figure 2. Interactions between target vehicle and other vehicles.

should also be taken into account. The interactions between those vehicles can be seen as a graph that contains nodes and edges. Nodes represent vehicles states (GPS location, speed, etc.) and edges represent the interactions (distance, angle, etc.) among them. Graph neural networks (GNNs) are a class of deep learning technologies designed to deal with graph data [16].

Cameras are also usually mounted on autonomous vehicles to capture traffic videos and even driver videos. They contain useful visual information for accurate vehicle maneuver prediction. Recent researches have applied attention models in detecting driver intentions and behaviors [25, 26, 19]. Human attention is a concept derived from human vision system where attention mechanism plays an essential role. It could be represented in a heat map generated from a front view video frame. A region in the heat map contains higher values if the corresponding region in the video frame attracts more attention from drivers during driving maneuvers. These areas contain the information useful in most drivers' decision process when making driving maneuvers. Attention models can be generated using deep learning techniques and large amounts of annotated video sequences of driving trips. Each frame was annotated with the regions where driver's gaze was registered. This inspired us to incorporate driver attention information into our system for detecting driving maneuvers.

Recently research has demonstrated that the driver's facial features could be used to understand and interpret driver's behaviors. However, in real-world driving trips, drivers' faces are not always visible, due to head motion, and object occlusion, such as hats, eye glasses, and etc. In this research we explore the use of drivers' facial features change as a surrogate feature for driver's head movement used in driver's maneuvering detection and classification. We will use a pre-trained model (Constrained Local Model) [37, 38] to locate the landmark points on driver's face, and then calculate the time differential facial features to characterize driver's head movements. Considering the situations in which the facial features are obscure, we developed a strategy for detecting the validity of the facial features.

In this dissertation, we propose novel deep learning-based approaches for vehicle maneuver prediction. Our vehicle maneuver prediction (VMP) systems utilize a combination of sensor data from cameras and vehicle signals to predict the future maneuver of a vehicle. The proposed VMP systems leverage the powerful representation learning capabilities of deep neural networks to capture complex patterns and relationships in the sensor data and predict the future maneuver of a vehicle.

Vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) and in-vehicle sensor data are all sources of information that can be used for vehicle maneuver prediction. We study the ego vehicle maneuver prediction based on information provided by in-vehicle sensors. The data contain ego vehicle GPS, ego vehicle speed, ego vehicle headings, front view video, and driver view video. We also study the remote target vehicle maneuver prediction using V2V data and V2I data. The V2V data contain the vehicle signals for both ego vehicle and remote vehicle, the V2I data have the vehicle signals for ego vehicle, remote target vehicle, and other surrounding vehicles. Fig. 3 shows those three kinds of data sources we will present in this research.

Each data source has its own advantages and disadvantages. V2V communication refers to the exchange of information between vehicles, which can include data such as position, speed, and heading. The use of V2V communication has a number of advantages and disadvantages. One of the main advantages of V2V communication is its ability to provide real-time, accurate information about nearby vehicles, which can improve the accuracy of vehicle maneuver prediction. However, V2V communication relies on the presence of other nearby vehicles that are equipped with V2V communication technology, which may not always be the case. Additionally,

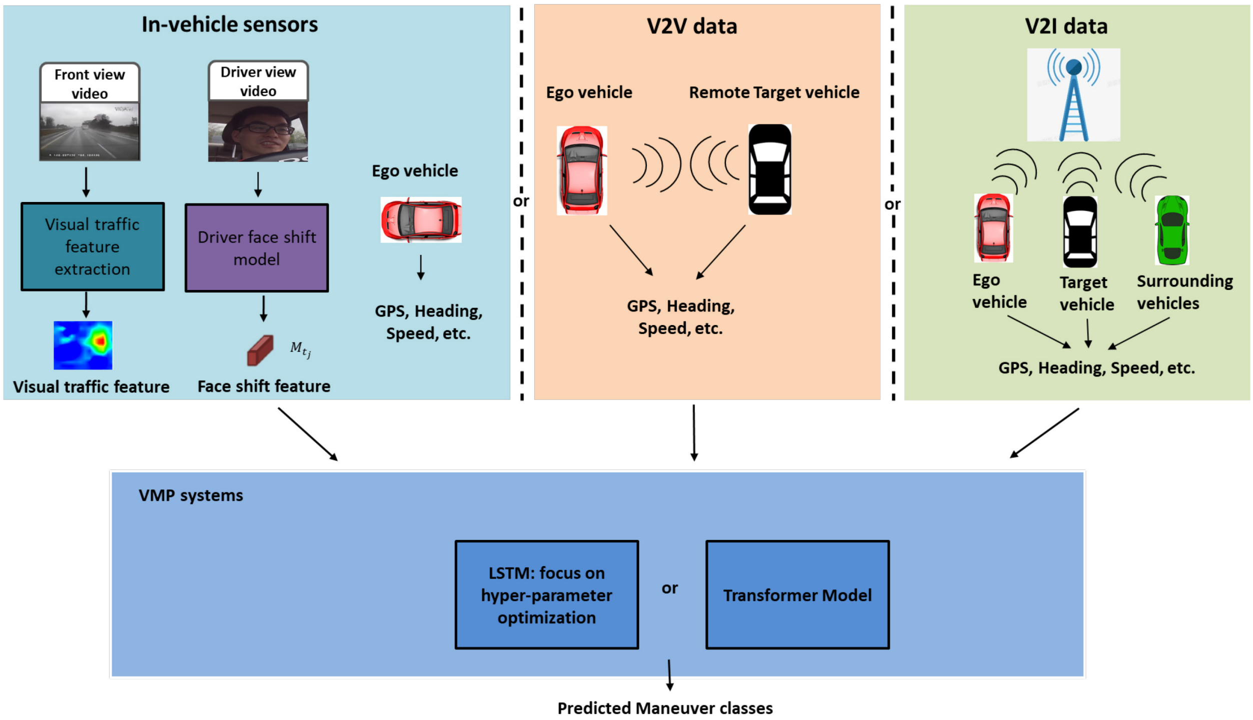


Figure 3. Our vehicle maneuver prediction research is based on in-vehicle sensors' data, V2V data, and V2I data.

V2V communication can be susceptible to communication delays or disruptions, which can impact the accuracy of the predictions. V2V communication involves the exchange of information between vehicles, which also can raise concerns about privacy and data security. V2I communication allows vehicles to receive information from infrastructure devices, such as traffic signals or road sensors. This information can include traffic conditions, road hazards, and other relevant information that can help a vehicle to make safer driving decisions. One of the main advantages of V2I communication is its ability to provide information about the road ahead that may not be visible to the driver, such as traffic congestion or road closures. However, V2I communication requires the installation of infrastructure devices, which can be costly and time-consuming to deploy. Additionally, the accuracy and reliability of V2I communication may be impacted by factors such as weather conditions or signal interference. In-vehicle sensors, such as cameras, lidar, or radar, provide information about the surrounding environment that can be used to predict vehicle maneuvers. These sensors can detect other vehicles, pedestrians, and road features, and can provide accurate information even in situations where V2V or V2I communication is not available. One of the main advantages of in-vehicle sensors is their ability to provide accurate information about the immediate surroundings of the vehicle. In-vehicle sensors do not rely on the presence of other nearby vehicles or infrastructure devices, which can make them more reliable in certain situations. However, in-vehicle sensors may be limited by factors such as weather conditions or sensor occlusion, which can impact their accuracy. In-vehicle sensors also have a limited range and may not be able to detect objects that are far away or outside of the sensor's field of view. And in-vehicle sensors require regular maintenance and calibration to ensure their accuracy and reliability, which can be time-consuming and costly.

The main contribution of this dissertation research is to provide a comprehensive solution for vehicle maneuver prediction that is robust and accurate. The proposed methods are evaluated on large-scale real-world datasets and their results are compared with several state-of-the-art approaches. Our results demonstrate the superiority of the proposed method in terms of prediction performance and interpretability.

1.2 Problem Formulation

The objective of this research is to predict a driving maneuver through an observed driving context.

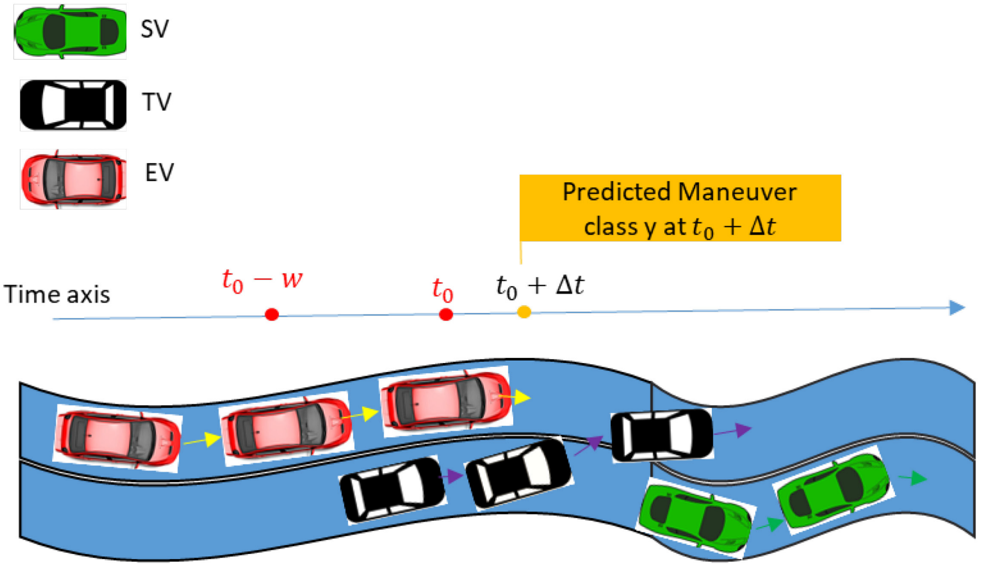


Figure 4. Illustration of vehicle maneuver prediction.

As illustrated in Fig. 4, all available observations at time t_0 to ego vehicle (EV) are $\boldsymbol{S} = \{X_{EV}, X_{TV}, X_{SV}\}$. Ego vehicle is the vehicle that equipped with this maneuver prediction system to predict target vehicle's maneuver. Target vehicle (TV) is the vehicle whose maneuver will be predicted in this research. The ego vehicle and the target vehicle could be the same vehicle when we want to predict ego vehicle's maneuver. And other surrounding traffic vehicles (SV) within 20 meters of ego vehicles will also be considered by the system.

- $X_{EV} = [x_{t_0-(w-1)}^{EV}, \dots, x_{t_0-1}^{EV}, x_{t_0}^{EV}]$, a sequence of driving states of ego vehicle from current time t_0 to past w seconds. And x_{t_0} consists of GPS location, speed, acceleration and direction of ego vehicle at time t_0.

- $X_{TV} = [x_{t_0-(w-1)}^{TV},\ \dots, x_{t_0-1}^{TV},\ x_{t_0}^{TV}]$, a sequence of driving states of target vehicle from current time t_0 to past *w* seconds.
- $X_{SV} = [x_{t_0-(w-1)}^{n}, \dots, x_{t_0-1}^{n}, x_{t_0}^{n}]_{n=1}^{N}$, a sequence of driving states of other *N* surrounding vehicles from current time t_0 to past *w* seconds.

At the current time t_0, the system need to predict target vehicle's maneuver class *y* occurred at time $t_0 + \Delta t$, where p is the look-ahead of time. Then the problem is defined as computing the conditional distribution $\boldsymbol{P}$ (y | S). In this research, we are interested in 5 maneuver classes $\boldsymbol{y} \in$ {left turn, right turn, left lane change, right lane change, going straight}.

1.3 Overview of the Dissertation Structure

This dissertation research will investigate VMP systems using LSTMs, Transformer and GNNs. The system will learn temporal features using LSTMs and Transformer through the current states and past states of vehicles (chapter 2). Those learnt temporal information of vehicles will be used by the system to make accurate predictions. I will also study how the visual information boost maneuver predictions in chapter 3. Chapter 4 will introduce how to use V2V data to predict the position of remote vehicle. The conclusion and future work will be discussed in chapter 5.

Chapter 2: Learning Temporal Dependencies in Driving Maneuvering

2.1 Introduction

The time series data learning models have proven to be effective in prediction of host and remote vehicle driving maneuvering. Learning temporal dependencies in driving states involves using machine learning algorithms to analyze driving data and identify patterns in the changes of driving states over time. These algorithms can learn to recognize and predict the changes in driving behaviors that occur as a driver navigates different road conditions, traffic patterns, and environmental factors. By identifying these temporal dependencies, machine learning algorithms can help to improve the accuracy of predictions of driving behaviors. Recurrent neural network (RNN) and Transformer are two of the widely used models in this area through different mechanisms.

LSTM is a special variant of Recurrent neural network (RNN) [18] that has achieved great successes in various applications in Natural Language Processing (NLP), Computer Vision (CV) and maneuver prediction applications due to its capability of learning long-term dependencies [19], [20]. This is because LSTM has unique architecture that is designed to capture long-term dependencies between input sequences. LSTM networks include specialized memory cells that can store and retrieve information over long time periods. This allows the network to remember past inputs and their associated context when processing current inputs, which is particularly useful in applications that require an understanding of context over time. LSTM networks also include input, output, and forget gates that control the flow of information through the network.

These gates allow the network to selectively remember or forget past inputs, which is crucial for dealing with noisy or irrelevant data in sequential data. By selectively retaining or discarding information, the network can focus on the most important features of the input sequence, leading to more accurate predictions.

The Transformer model is a type of neural network architecture that was originally designed for natural language processing tasks, such as language translation and text generation. However, it has also been found to be effective in learning temporal dependencies in sequential data, including in the context of vehicle maneuver prediction.

The Transformer model is capable of learning temporal dependencies in sequential data due to its use of attention mechanisms, which allow the model to focus on relevant features at different time steps. Unlike recurrent neural networks, which process input sequences in a sequential manner, the Transformer model can process all input data simultaneously, making it more efficient and less prone to vanishing gradients. Attention mechanisms in the Transformer model allow it to assign different weights to different parts of the input sequence, allowing it to selectively focus on the most important features of the data. This is particularly useful in vehicle maneuver prediction applications, where relevant features may occur at different time steps and where identifying the most important features is crucial.

Additionally, the Transformer model includes multiple layers of self-attention, which allow it to learn complex, hierarchical representations of the input sequence. By recursively attending to the most relevant features of the input sequence, the Transformer model can learn increasingly abstract and sophisticated representations, leading to more accurate predictions.

Both LSTM and Transformer models can be effective in capturing long-term dependencies and identifying the most important features of the input sequence, making them suitable for applications such as vehicle maneuver prediction.

In this chapter, we show that the accuracy of VMP can be significantly improved through learning temporal dependencies in driving states using LSTM and Transformer based models. The problem can be solved with an end-to-end machine learning architecture, where the data from different sensors are fused by a LSTM/Transformer network. The proposed LSTM/Transformer based VMP systems outperform models that do not consider temporal dependencies.

The rest of the chapter is organized as follows. Section 3.2 summarizes the related literature. In Section 3.3, we describe LSTM and Transformer network are designed to learn the temporal information in sequential data. We present the design and the results of empirical experiments in Section 3.4 and Section 3.5, and then conclude the chapter in Section 3.6.

2.2 Related Work

2.2.1 Recurrent Neural Networks

Many of the driving maneuver detection and prediction systems used LSTM and Gated recurrent unit (GRU) [41] to learn useful knowledge from sequences of vehicle driving data. Authors in [8] developed a deep learning algorithm for learning the relationship between driving maneuvers and traffic scenes captured by the front view video camera. The algorithm used both the InceptionV2 features with transfer learning from a pre-trained CNN model [9], and the vehicle signals as the input to a LSTM network. The model was trained to detect 11 different driving maneuver classes of traffic vehicles including turns, lane change, intersection passing, etc. They evaluated their system by using the Honda research institute Driving Dataset, which contains 104 hours of real driving trips, which recorded GPS, LiDAR and front view videos via an instrumented vehicle. They demonstrated that their proposed algorithm outperformed the other 4 baseline models (random guessing, CNN pooling + LSTM, vehicle signals + LSTM, convolutional net + LSTM) with 32.71 mean Average Precision (mAP).

Xu et al. [10] proposed an end-to-end FCN-LSTM network to predict driving maneuvers in an ego-vehicle. The model contained a visual encoder to extract visual representations from the front view videos with a fine-tuned AlexNet model, and then used the fused temporal visual features and sensor signals as input to the LSTM model to learn temporal information. The system was evaluated using a set of 300-hours of real world driving data extracted from the BDDV (Berkeley Deep Drive Video) dataset. It reached an accuracy of 84.1% in predicting four maneuvering classes, going straight, and stop, left and right turn. However, the paper did not provide the details of the number of data samples in each of the four maneuvering categories, and the prediction results in each maneuvering category. However, the study did not include the maneuver classes of "left and right lane change", which are very important information for an ADAS, since many accidents happen during these two types of maneuvering.

Meanwhile, [40] proposed a two-part approach, wherein two separate LSTM networks were utilized to learn different features: one for driver-based features from in-cabin video and another for surrounding-based features from external vehicle video. One innovation point is that they developed a facial landmark point detector was developed for tracking landmark points on driver's face, and then optical flow trajectories were generated from these fixed points. Furthermore, they projected 2D landmark points to 3D head model to estimate three head poses of yaw, pitch and row. These new features were combined with vehicle signals for training a maneuver prediction model. They showed that the system used these new features performed 6% higher in precision better than the model without these features.

The paper [42] presents a lane change maneuver detection system that utilizes Gated Recurrent Units (GRUs) to model pairwise interactions between a target vehicle (RV) and adjacent surrounding vehicles (SVs). The input data consists of 1669 lane change maneuvers extracted from the NGSIM dataset, with the history of states including GPS, speed, headings, and distance to current lane centers of the RV and four adjacent SVs from the last 2 seconds. The output of the system is the predicted lane keeping, lane change left, and lane change right of the RV in the next 4 seconds in highway driving scenarios. The authors use a group of GRUs to model the pairwise interactions between the RV and each of the adjacent SVs. The system achieves a high F1 score of 94.4%, demonstrating the effectiveness of the proposed approach.

Ma et al. [43] presented an LSTM-based traffic prediction algorithm called TrafficPredict that operates in heterogeneous traffic, i.e. containing vehicles and pedestrians. TrafficPredict was constructed using a 4D Graph, which can be divided into two main layers: the instance layer and the category layer. The instance layer was designed to capture dynamic properties and interactions among traffic-agents at a micro level. The category layer was designed to learn the behavior similarities of instances of the same category using a macroscopic view and guide the prediction for instances. The category layer also used a self-attention mechanism to capture historical movement patterns and highlight category differences. The authors evaluated TrafficPredict by using a dataset collected on urban streets by a car equipped with a variety of sensors, including LiDAR (Velodyne HDL-64E S3), radar (Continental ARS408-21), camera, and high-definition maps. Experimental results showed that TrafficPredict outperforms other state-of-the-art approaches.

LSTM has also been used with convolutional neural networks (CNN). In [44], the authors proposed a Multi-Agent Tensor Fusion (MATF) network to models multiple agent past trajectories and the scene context into a Multi-Agent Tensor. There are two parallel encoding streams in the MATF architecture. One encodes the past trajectories of each individual agent independently using LSTM encoders, and another encodes the static scene context image using a CNN. In [39], the authors used a graph to represent the interactions of close objects, and a LSTM encoder-decoder network to predict the future trajectories of traffic agents around an autonomous car. The LSTM encoder-decoder network took the computed output of a graph convolutional model as input. Then, the hidden features of the encoder LSTM, as well as coordinates of objects at the previous time step, were fed into a decoder LSTM to predict the position coordinates at the current time step.

2.2.2 Transformer Models

While the transformer network models were first popular for natural language processing tasks [45], its application to sequential event classification and prediction problems is just starting to be explored. In computer vision, transformers have been mainly used in conjunction with convolutional networks, or used to replace certain components of convolutional networks, while keeping the overall CNN structure in place. These systems are often pre-trained on large amounts of data and then transferred to various mid-sized or small image recognition benchmarks. The Vision Transformer (ViT) model presented in [46] was the first that applied a pure transformer directly to sequences of image patches and performed very well on image classification tasks compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train. The ViT transformer receives a 1D input sequence of flattened 2D patches. The transformer uses a constant latent vector size D through all of its layers with a trainable linear projection. The authors demonstrated that the ViT is scalable and works well on object recognition tasks when coupled with pre-training on large datasets.

More recently, transformer networks have also been investigated in the application of predicting vehicle [47] and pedestrian [48] trajectories in urban scenes. The authors in [47] and [48] built on the TF architecture in [45] to create a system consisting of an encoder stage and a decoder stage, where both encoder and decoder are composed of 6 layers, and each layer contains three building blocks: an attention module, a feed-forward fully-connected module, and two residual connections after each of the previous blocks. The attention modules are developed to capture sequence nonlinearities.

In [47], the authors used the same TF network presented in [45] to predict vehicle trajectory in urban scenarios in prediction horizons up to 5 seconds. The input to the TF network was a sequence of vehicle coordinates and vehicle heading within an observation window in the time domain. The input values are all normalized using the z-score method, where the mean and standard deviation were obtained from the training data. In order to incorporate temporal information into the input embedding, the corresponding timestamp is transformed through sine and cosine functions. The output is the predicted vehicle coordinates. The TF system has the same architecture as the one presented in [48]. The authors evaluated the TF network on data sets that contain traffics in urban intersections, roundabouts, and highways. The results showed that the TF systems gave competitive performances over among the state-of-the-art models including LSTM models.

In [48], the authors investigated using both the original TF network and the larger Bidirectional Transformer (BERT) network to predict the trajectories of pedestrians in a scene. For each person, the transformer network outputs the predicted future positions by processing their current and prior positions (observations or motion history as represented by Cartesian coordinates). These are “simple” models because each person is modelled separately without any complex human-human nor scene interaction terms. BERT is the de-facto reference model for state-of-the-art NLP methods, but usually larger than TF-based models (~2.2 times larger). However, the authors showed that training BERT on the current largest trajectory forecasting benchmarks produces state-of-the-art performance. The BERT performance may indicate that the model does require a much larger amount of training data, which is not as plentiful as a typical NLP dataset. When tested on a variety of pedestrian trajectory data sets, the proposed TF-based system achieved the best performance when compared to 40 other state-of-art techniques, including LSTM-based networks.

2.3 VMP Systems Using Deep Sequential Learning Algorithm

The VMP systems are multi-layer networks that receive multi-modal features from different vehicles (target vehicle, ego vehicle, and other vehicles). The networks will learn temporal dependencies in feathers and fuse them together. Driving maneuver prediction could be seen as finding a target maneuvering class, ***y,*** that has the maximum probability of driver's maneuvering intension given the vehicle information and driving context ***S***.

2.3.1 LSTM Network based VMP system

In this subsection we present the LSTM-based models we developed to predict vehicle maneuver from driving states of TV and SVs. The system is demoted as VMP-L.

Let $\mathbf{z}_t \in R^n$ denote the vehicle features extracted at time *t*. The input to both the LSTM and transformer-based DLNNs is a temporal feature vector $\mathbf{Z}_t \in R^{n\times w}$ that consists of features extracted from a sliding observation window of size *w:*

$$\mathbf{Z}_t = [\,\mathbf{z}_{t-(w-1)} | \dots \;|\; \mathbf{z}_{t-2} \;|\; \mathbf{z}_{t-1} \;|\; \mathbf{z}_t\;]$$

The structure of the proposed LSTM network is shown in Figure 5. Here, there are L LSTM layers, followed by a fully connected layer. The overall task of the network is to map the $n \times w$ input to a *K*-dimensional 1-hot output vector $\hat{\mathbf{y}}_{t+\Delta t}$, which represents the output class (after softmax).

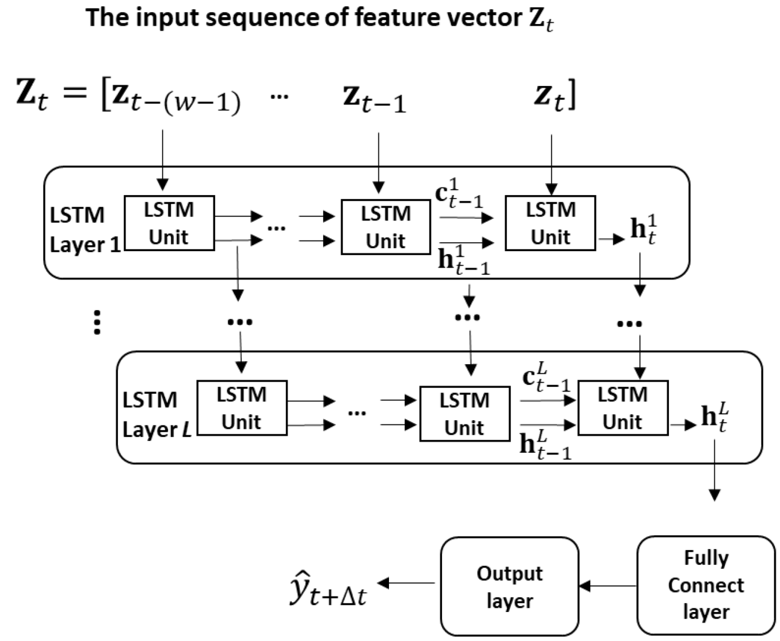


Figure 5. Architecture of a multilayered LSTM network.

Let $\mathbf{h}_t^l$ denote the output of the lth layer at time *t*, where $l \in \{1, 2, \dots, L\}$. The essential mechanism of LSTMs is that each unit can remember its state over time as shown in Fig. 6. The key components are three gates: *input gate* i_t, *output gate* o_t, and *forget gate* f_t, and a *memory cell* c_t. The input gate decides what information can be added into the cell, forget gate resets the content of the cell, and output gate reads out the entries from the cell. At each time step *t*, LSTMs update the input and forget gate activations i.e. $\mathbf{i}_t$ and $\mathbf{f}_t$ based on the input feature vector $\mathbf{z}_t$ and the previous hidden state $\mathbf{h}_{t-1}$. For notional simplicity, we suppress the layer l superscript. The update equations for $\mathbf{i}_t$ and $\mathbf{f}_t$ are given by:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{zi}\mathbf{z}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i) \quad (2)$$

$$\mathbf{f}_t = \sigma\big(\mathbf{W}_{zf}\mathbf{z}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f\big) \quad (3)$$

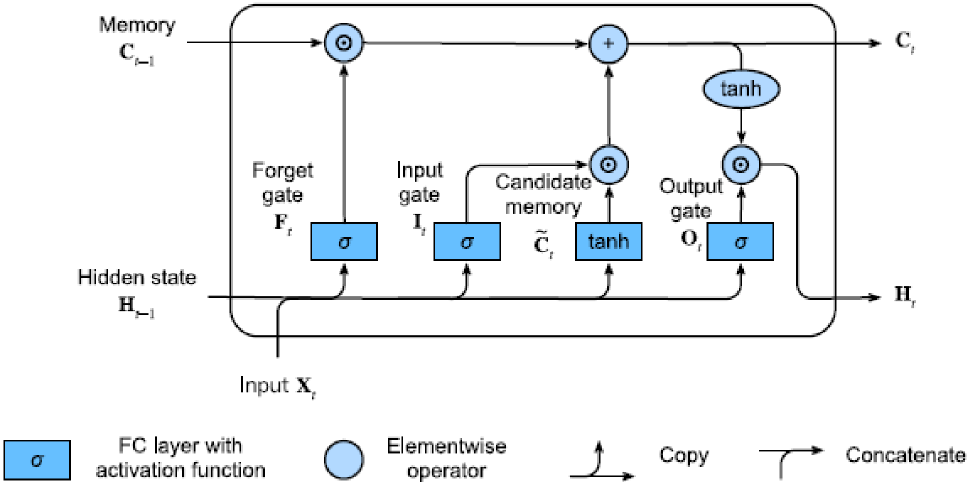


Figure 6. LSTM cell.

where $\sigma(x)$ is the sigmoid activation function, and $\mathbf{W}_{\alpha,\beta}$ and $\mathbf{b}_{\beta}$ are the weight matrix and bias term that are used to take α as input and produce β as output through (2) to (6), where $\alpha \in \{z, h, c\}$, $\beta \in \{i, f, c, o\}$.

The memory cell is then updated from $\mathbf{c}_{t-1}$ to $\mathbf{c}_t$ using

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_{zc}\mathbf{z}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \quad (4)$$

Finally, the output gate activation $\mathbf{o}_t$ and the hidden representation $\mathbf{h}_t$ are computed using:

$$\mathbf{o}_t = \sigma(\mathbf{W}_{zo}\mathbf{z}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o) \qquad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \qquad (6)$$

where ⊙ denotes an element-wise product.

In Fig. 5, the fully connected layer maps the hidden vectors $\mathbf{h}_t$ from the last layer to the *K*-dimensional output $\hat{\mathbf{y}}_{t+\Delta t}$, where $\Delta t > 0$. A softmax function is used to estimate the probability of each of the output classes at time $t+\Delta t$, and finally, $\hat{\mathbf{y}}_{t+\Delta t}$ is given by:

$$\mathbf{p}_{t+\Delta t} = softmax\left(FC(\mathbf{h}_t)\right) \qquad (7)$$

$$\hat{\mathbf{y}}_{t+\Delta t} = \arg max\ (\mathbf{p}_{t+\Delta t}) \qquad (8)$$

Various loss functions can be used in the training process to update the network weights. In this work, we used the cross entropy loss function:

$$L = -\sum_{i=1}^{m}\sum_{k=1}^{K} I_{ik} \ln\ (\mathbf{p}_{ik}) \qquad (9)$$

where *M* is the number of samples in the training data, *K* is the number of output classes, I_{ik} is an indicator function where $I_{ik} = 1$ if sample i belongs to class k, and 0 otherwise. And p_{ik} is the computed probability that sample *i* belongs to class *k.*

2.3.2 Transformer Network based VMP system

a. Softmax function

Softmax operation is a mathematical function that is often used in machine learning and deep learning models to convert a vector of arbitrary real values into a probability distribution over the same set of values. The softmax function is defined as:

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_n x_j} \text{ for j in 1 to n.}$$

where x_i is the i_{th} element of the input vector, and ***n*** is the total number of elements in the vector. The softmax function exponentiates each element of the input vector and divides it by the sum of all exponentiated values, resulting in a set of values that sum to 1 and can be interpreted as probabilities.

The softmax operation is commonly used in classification tasks, where the goal is to predict the probability of an input belonging to each possible class. For example, in an image recognition task, the softmax function could be used to predict the probability that the input image belongs to each of several categories (e.g., dog, cat, bird).

The softmax function is differentiable, which makes it suitable for use in deep learning models that use backpropagation for optimization. The output of the softmax operation can be used as the final output of a model, or as an intermediate step in a more complex computation.

b. Self-attention mechanism

Self-attention is a mechanism commonly used in deep learning models, particularly in natural language processing and computer vision tasks. Self-attention allows the model to weigh different parts of the input data, based on their relevance to the output, by computing the similarity between all pairs of positions in the input sequence. This allows the model to attend to different parts of the input data, depending on the context and the task at hand. In this mechanism, the input sequence is first transformed into three sets of vectors: **Query**, **Key**, and **Value** vectors. The query vector represents the current position in the sequence, while the key and value vectors represent all other positions in the sequence. Next, the dot product of the query vector with all key vectors in the sequence is calculated, resulting in a set of scores. These scores are then normalized using a softmax function, which produces a set of attention weights that represent the importance of each position in the sequence for the task at hand. Finally, the attention weights are used to compute a weighted sum of the value vectors, where each value vector is multiplied by its corresponding attention weight. The resulting vector represents the context vector, which summarizes the most important parts of the input sequence for the task at hand. Since the queries, keys, and values come from the same place, this performs self-attention, which is also called intra-attention.

$$\text{Attention output} = \text{softmax}(\frac{\mathbf{QueryKey}^{T}}{\sqrt{d}})\mathbf{Value}$$

where ***d*** is the length of **Query** and **Key**, *T* is transpose operation.

The use of self-attention can be beneficial in several ways. Firstly, it allows the model to capture complex relationships between different parts of the input data, which can be difficult to achieve using traditional convolutional or recurrent neural network architectures. This can improve the performance and accuracy of the model, especially in tasks that require a deep understanding of the input data, such as vehicle maneuver prediction.

c. Multi-head attention

Multi-head attention is an extension of the self-attention mechanism used in deep learning models. Multi-head attention allows a model to attend to multiple parts of the input sequence simultaneously, enabling it to capture more complex relationships and dependencies between different parts of the sequence.

Instead of performing a single attention pooling, multi-head attention uses K independently learned linear projections to transform the queries, keys, and values. These K projected vectors are then fed into attention pooling in parallel, producing K attention pooling outputs, also known as "heads". The K attention pooling outputs are concatenated and transformed with another learned linear projection to produce the final output. This approach allows the model to attend to multiple aspects of the input sequence simultaneously, resulting in a more sophisticated representation of the input. The use of fully-connected layers to perform learnable linear transformations is depicted in Figure 7.

By using multiple heads, the model can learn to attend to different aspects of the input sequence and combine the information in a more sophisticated way. This approach has been shown to improve the performance of deep learning models on a variety of tasks, including natural language processing, speech recognition, and image recognition.

Multi-head attention is used in many state-of-the-art deep learning models, including BERT, GPT-2, and Transformer, and has become a standard building block in many architectures.

d. Our proposed attention mechanism based system

We also developed a VMP system based on transformer networks (VMP-T). The system is modeled after the ViT architecture presented in [46], and is built with a self-attention mechanism; i.e. it does not contain any convolutional or recurrent layers.

The overall architecture of VMP-T is presented in Fig. 7. The system consists of a feature embedding layer, an encoder, and a classification layer. The input to the VMP-T system is the feature vector $\mathbf{z}_t$ over an observation window of size w (same as the LSTM model).

The transformer encoder is a stack of identical encoder blocks, and each contains a multi-head self-attention (MSA) layer and a position-wise feed-forward neural network (FNN). Specifically, in the encoder self-attention, the inputs are taken from the outputs of the previous encoder block, except the first one, which takes the output from the embedding layer, denoted as $\mathbf{g}$. A residual connection is applied to both MAS and FNN. Finally, a classification layer is connected to the last encoder block to generate predicted classes.

In self-attention, $\mathbf{g}$ is sent to a multi-head attention layer to learn different patterns contained in the input sequence, such as capturing dependencies between different positions in the sequence.

Then an embedding layer takes input sequence $\mathbf{z}$ as input and outputs generate a embedding feature . Then $\mathbf{g}$ passes by each of encoder blocks and the output from the lth encoder is denoted as $\mathbf{g}_l$, which is used as input to the encoder at layer $l+1$, $l \in \{1, 2, \ldots, L\}$. Finally, the outputs from the attention pooling layers are concatenated and fed into another FC layer to generate the final output. The computational steps in the $\boldsymbol{l_{th}}$ transformer encoder can be formulated as:

$$\mathbf{g}_l' = MSA(\mathbf{g}_{l-1}) + \mathbf{g}_{l-1}\,, \qquad l = 1, \ldots, L \qquad (10)$$

$$\mathbf{g}_l = FC\big(norm(\mathbf{g}_l')\big) + \mathbf{g}_l'\,, \qquad l = 1, \ldots, L \qquad (11)$$

$$\hat{\mathbf{y}}_{t+\Delta t} = FC(\mathbf{g}_L) \qquad (12)$$

2.4 Experiment Setup

2.4.1 Datasets

Next Generation Simulation (NGSIM) dataset. The most commonly used dataset in the literature for vehicle maneuver prediction is the Next Generation Simulation (NGSIM) public dataset [30]. This dataset contains actual driving data that can be used to predict the future behavior of a vehicle in relation to neighboring vehicles. The NGSIM dataset has been used in several related research studies [43, 44, 39].

The dataset was collected using Next Generation Simulation (NGSIM) software in 4 different locations: southbound US 101 and Lankershim Boulevard in Los Angeles, California, eastbound I-80 in Emeryville, California; and Peachtree Street in Atlanta, Georgia. A network of synchronized digital video cameras was used to capture comprehensive vehicle trajectory data. After capturing the video footage, the vehicle trajectory data was extracted using a specialized software application called NGVIDEO, developed specifically for the NGSIM program. This

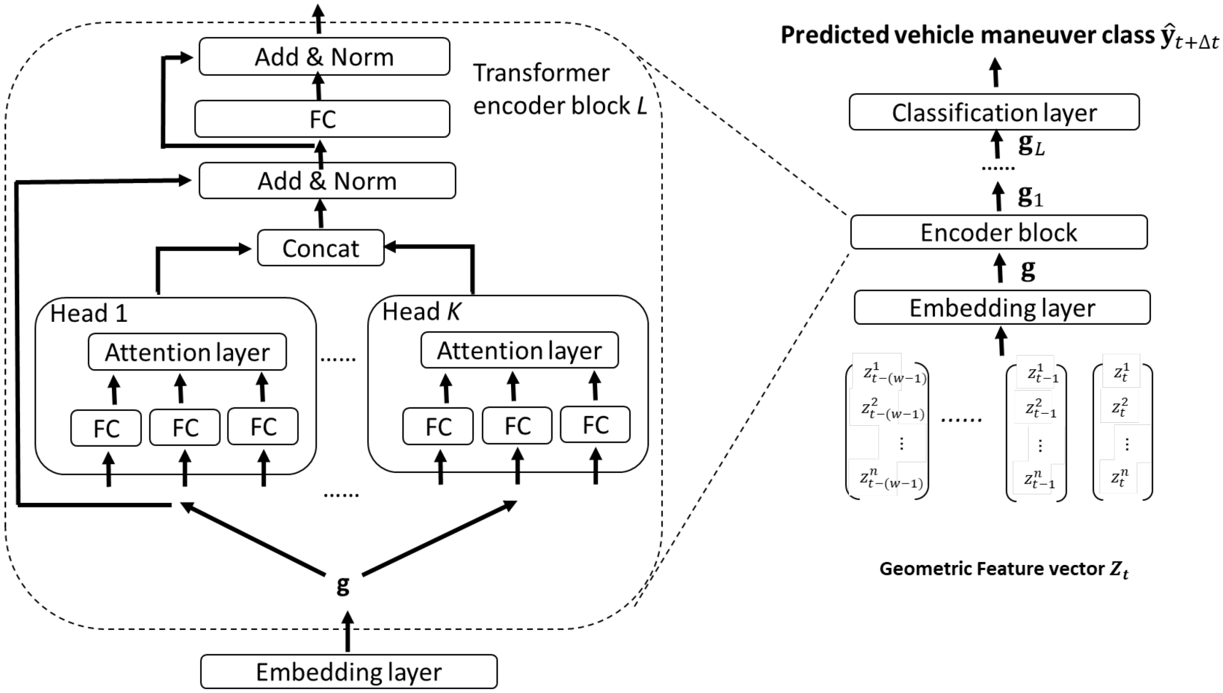


Figure 7. VMP-T: an attention focused transformer network for vehicle maneuver prediction.

resulted in data that provided the exact location of each vehicle within the study area every one-tenth of a second, resulting in detailed information on lane positions and the location of neighboring vehicles. The duration of dataset is over 45 minutes and the data were collected with 10 Hz. The dataset has 25 attributes, includes vehicle ID, timestamp, Lateral coordinate, Longitudinal coordinate, velocity, acceleration, lane ID, direction, etc. Their maneuvers were annotated as going straight, left turn, and right turn. But the lane change maneuver can be inferred from lane ID. Table 2.1 gives statistics of vehicle data in 4 different locations. Fig. 8 shows an example that EV makes a left lane change maneuver in 5 seconds, the locations of EV and 4 SVs are presented by dots in 1 Hz.

Table 2.1 Vehicles and trip durations in each location in NGSIM dataset

Location	#vehicles	Average trip length
US-101	2847	65.4s
I-80	1725	60.7s
Peachtree Street	1862	57.6s
Lankershim Boulevard	1379	49.8s

The raw NGSIM trajectory data contain the attributes of vehicle ID, Lane ID, Speed, acceleration, GPS, and direction. We aim to extract trip sequences of 5 driving maneuver classes. They are left turns and right turns, left lane change, right lane change and going straight. The maneuver of turns was already labeled in the dataset, so we need to find out the trips with lane changes to extract them.

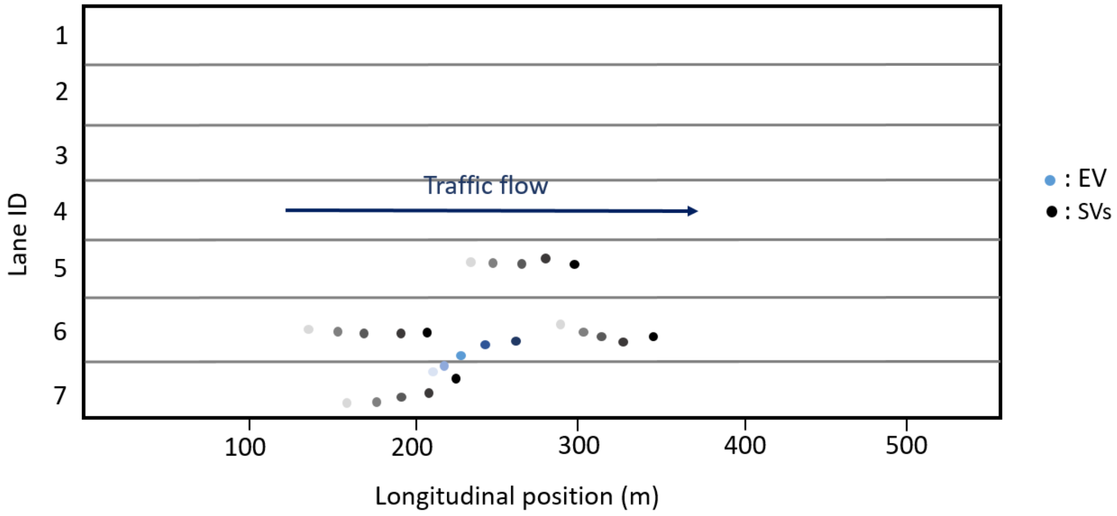


Figure 8. Traffic flow of EV and 4 SVs in 5 seconds on I-80 highway.

Fig. 9 shows the workflow of lane change maneuver. The lane change maneuvers can be extracted by finding the changes of lane ID. For a lane change maneuver occurred at $t + \Delta t$, extract the data point *x* between [*t-(w-1), t*] for TV and N SVs.

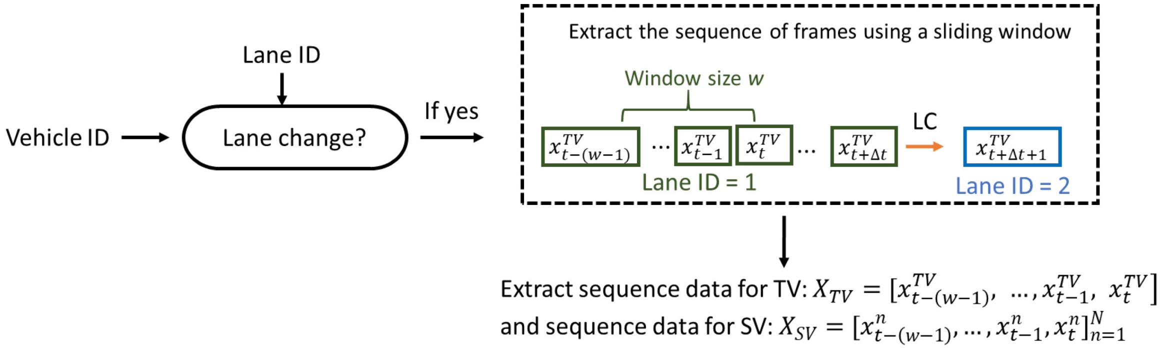

Lane ID
Vehicle ID
Lane change?
If yes
Extract the sequence of frames using a sliding window
Window size w
$x_{t-(w-1)}^{TV}$
x_{t-1}^{TV}
x_{t}^{TV}
$x_{t+\Delta t}^{TV}$
LC
$x_{t+\Delta t+1}^{TV}$
Lane ID = 1
Lane ID = 2
Extract sequence data for TV: $X_{TV} = [x_{t-(w-1)}^{TV}, \ldots, x_{t-1}^{TV}, x_{t}^{TV}]$
and sequence data for SV: $X_{SV} = [x_{t-(w-1)}^{n}, \ldots, x_{t-1}^{n}, x_{t}^{n}]_{n=1}^{N}$

Figure 9. Workflow of sequence data extraction for lane change maneuver.

In NGSIM dataset, all vehicle data were captured through V2I communication and no EV in this situation. Thus the data $\mathbf{Z}_t$ is shown in Eq. (13) and maneuver class at $t\ +\Delta t$ is $y_{t\ +\Delta t}\ \in$ $\{\text{Going straight, left turn, right turn, left lane change, right lane change}\}$.

$$\mathbf{Z}_t = [\ \mathbf{z}_{t-(w-1)}| \ldots\ \ |\ \ \mathbf{z}_{t-2}\ \ |\ \ \mathbf{z}_{t-1}\,|\,\mathbf{z}_t\ \ ]$$

$$= \begin{bmatrix} x_{t-(w-1)}^{TV} & \cdots & x_{t-1}^{TV} & x_{t}^{TV} \\ x_{t-(w-1)}^{SV} & \cdots & x_{t-1}^{SV} & x_{t}^{SV} \end{bmatrix} \qquad (13)$$

I choose this dataset because it provides sufficient vehicle driving states in time series, this allows this research to study the temporal and spatial features. NGSIM dataset is also a large benchmark dataset in vehicle maneuver predicting area and has been widely used in many other works. This is helpful in evaluating my system by comparing performances with others. The inadequacy of the dataset is that it lack of video data of front view. Fig. 10 shows the workflow if using V2I data in VMP systems.

UMD-ISL dataset. ISL-UMD dataset contains video sequences of real-world driving trips taken by 20 different drivers, with the total time of 107 hours and 4568 km in distance. The ISL-UMD dataset were collected and annotated by members in the Intelligence Systems Lab. Each trip contains various vehicle signals including the vehicle positions measured in GPS, vehicle speed

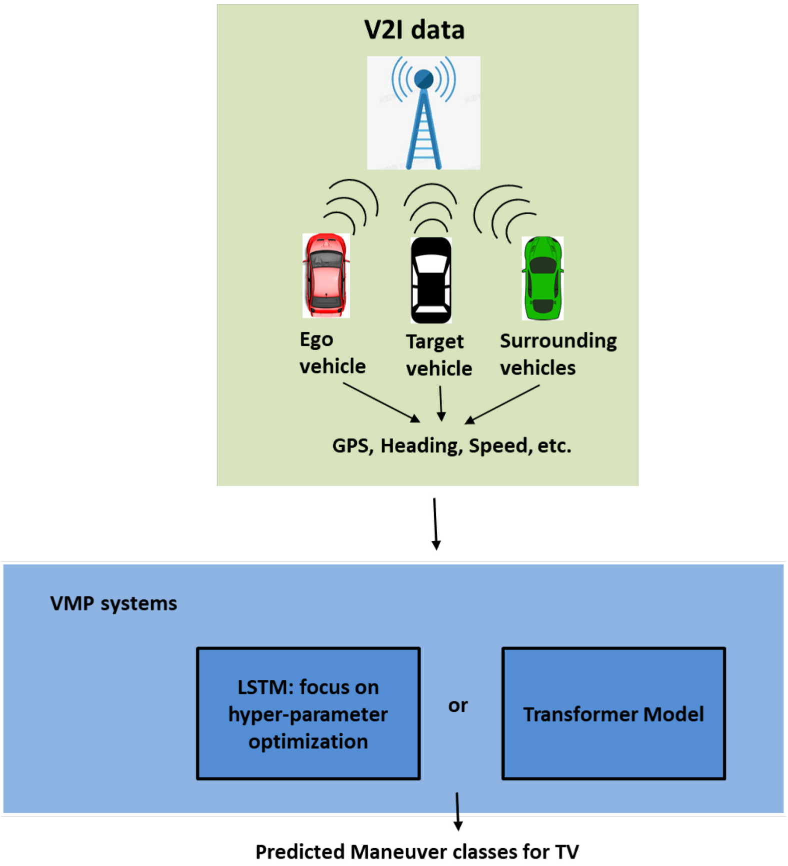

V2I data
Ego vehicle
Target vehicle
Surrounding vehicles
GPS, Heading, Speed, etc.
VMP systems
LSTM: focus on hyper-parameter optimization
or
Transformer Model
Predicted Maneuver classes for TV

Figure 10. Target vehicle maneuver prediction using V2I data in NGSIM dataset.

and heading, and driving videos taken by two cameras, which provides a front road view and a driver view. The dataset is challenging due to the dynamics of the real-world traffic environment, large variations in light conditions and the individual driver's styles and behaviors.

The data are annotated in five maneuvering classes, left and right turns, left and right lane changes, and going-straight. All ISL-UMD data were manually labeled by independent annotators and verified by one expert for consistency. The statistics of the five classes of driving maneuvering data samples in ISL-UMD are shown in Fig. 11 (a). There are 3129 maneuvers are annotated in total, including 771 left turns, 788 right turns, 516 left lane changes, 485 right lane changes, and 569 going straight.

The UMD-ISL dataset contains images of front view and driver view, but it only has driving states of the EV. So in the experiments on UMD-ISL dataset, EV and TV will be the same vehicle, which means we will predict the maneuver of ego vehicle. Thus the data $\mathbf{Z}_t$ is shown in Eq. (14) and maneuver class at $t\ +\Delta t$ is $y_{t\ +\Delta t}\ \in$ { Going straight, left turn, right turn, left lane change, right lane change }.

$$\mathbf{Z}_t = [\ \mathbf{z}_{t-(w-1)}| \dots\ \mid\ \mathbf{z}_{t-2}\ \mid\ \mathbf{z}_{t-1} \mid \mathbf{z}_t\ ]$$

$$= \left[x_{t-(w-1)}^{TV}\quad \dots\quad x_{t-1}^{TV}\quad x_t^{TV}\right] \quad (14)$$

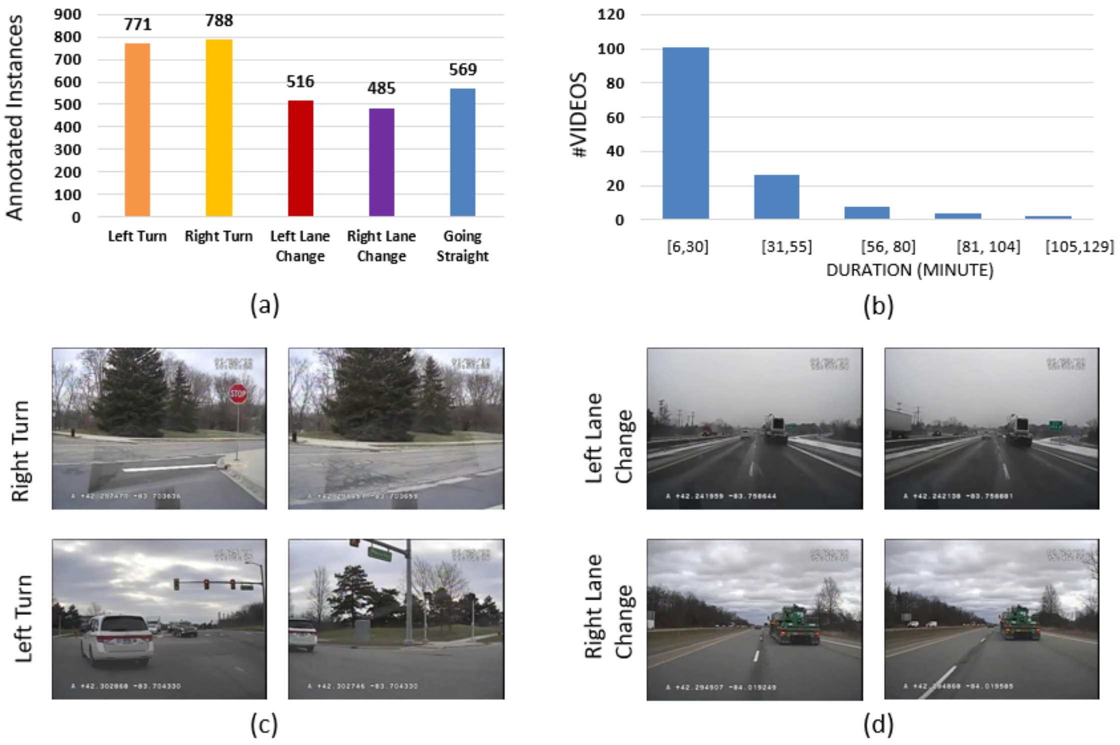


Figure 11. Statistics and examples of ISL-UMD dataset. (a) The number of samples in each of the five annotated maneuvering classes. (b) The distribution of trip duration. (c) Examples of four different driving maneuvering classes.

Fig 12 gives the workflow of ego vehicle maneuver prediction using in-vehicle sensors' data.

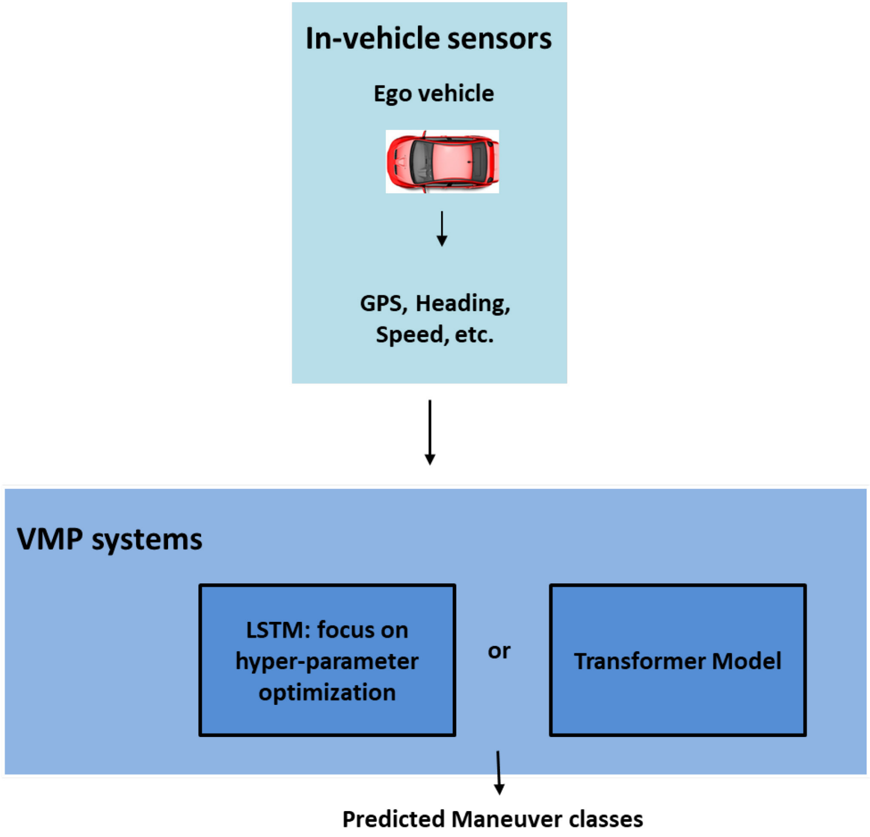

In-vehicle sensors
Ego vehicle
GPS, Heading, Speed, etc.
VMP systems
LSTM: focus on hyper-parameter optimization
or
Transformer Model
Predicted Maneuver classes

Figure 12. Ego vehicle maneuver prediction using in-vehicle sensors' data in UMD-ISL dataset.

2.4.2 Evaluation Metrics

We use recall rate, precision and F1 score as our evaluation metrics, which are common choices for classification tasks.

Recall rate measures the percentage of positive samples that were correctly predicted by the system. In other words, it measures the ability of the model to predict all relevant samples. Recall is calculated as the ratio of true positives to the sum of true positives and false negatives:

$$Recall = \frac{True\ positives}{True\ Positives\ +\ False\ Negatives}$$

Precision measures the percentage of predicted positive samples that are actually positive. In other words, it measures the ability of the model to correctly predict positive samples without including irrelevant samples. Precision is calculated as the ratio of true positives to the sum of true positives and false positives:

$$Precision\ = \frac{True\ positives}{True\ Positives\ +\ False\ Positives}$$

The F1 score is a combination of recall and precision that provides a single, balanced measure of a classification model's performance. The F1 score is calculated as the harmonic mean of recall and precision, and can be expressed as:

$$F1\ score = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

The F1 score ranges from 0 to 1, with a higher score indicating better performance. The F1 score is useful when both precision and recall are important, and provides a more comprehensive measure of a classification model's performance than either recall or precision alone.

2.4.3 Baseline Models

To illustrate the effectiveness of the proposed sequential learning model for VMP system, we compared VMP-L and VMP-T against with following model,

- VMP-NN: a neural network model that takes $\mathbf{z}_t$ at single time point *t* as input.

2.4.4 Hyper-parameter Settings in VMP-L and VMP-T

In the experiments in this chapter, we used the following hyper-parameter settings:

- Data sampling rate: 10 Hz
- Window size: $w = 5$ seconds

- Number of output classes: 5
- Max number of training epochs: 500
- Number of LSTM layers: 3
- Number of heads in VMP-T: 6
- Number of encoder blocks in VMP-T: 3
- Look ahead prediction time: Δt = 1, 2, 3, 4, 5 seconds
- Learning rate: 0.001
- Batch size = 64

In our experiments, NGSIM and UMD-ISL data were split into 70% training set and 30% validation set. With the above hyper-parameters, there are 77,509 learnable weights in VMP-T and 87,877 learnable weights in VMP-L when they were evaluated on NGSIM dataset. The systems trained on UMD-ISL dataset have 76741 and 84805 weights, because the input feature size decreased as we discussed. The training time for a VMP-T model was about 9 hours, while a VMP-L model took about 11 hours.

2.5 Experiment Results

We evaluated the performance of three vehicle maneuver prediction systems using the F1 score. The three systems utilized different deep learning neural network architectures, including a conventional neural network (VMP-NN), a LSTM network (VMP-L), and a transformer network (VMP-T). It should be noted that the vehicle maneuver prediction is made at time *t* for vehicle maneuver at time t+Δt. The F1 score was computed for each system at five different prediction horizons (Δt = 1s, 2s, 3s, 4s, and 5s). The results are summarized in the following tables 2.2 and 2.3.

As expected, the prediction results, in general, decrease as the prediction horizon increases. The VMP-T system gave the highest F1 score of 84.2% in comparison to the VMP-NN and VMP-L. The proposed sequential data learning systems, VMP-T and the VMP-L systems outperform the VMP-NN systems in prediction horizons by large margins. Additionally, the three systems got higher F1 score by 3%~7% on NGSIM dataset than on UMD-ISL data .

Table 2.2: F1 score for vehicle maneuver prediction on NGSIM dataset

System	Prediction Horizon Δt (s)				
	1	2	3	4	5
VMP-NN	65.2%	62.4%	57.5%	54.2%	52.9%
VMP-L	80.6%	77.2%	73.2%	71.0%	67.5%
VMP-T	84.2%	81.3%	78.4%	76.5%	75.7%

Table 2.3: F1 score for vehicle maneuver prediction on UMD-ISL dataset

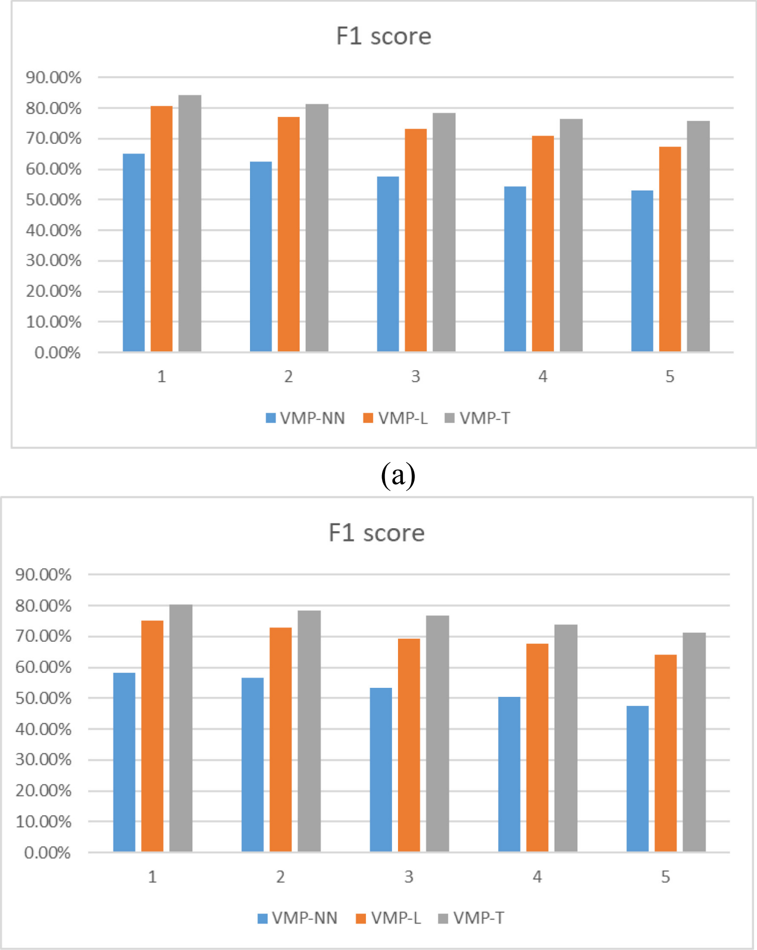
System	Prediction Horizon Δt (s)				
	1	2	3	4	5
VMP-NN	58.3%	56.6%	53.2%	50.4%	47.4%
VMP-L	75.1%	72.9%	69.4%	67.5%	64.0%
VMP-T	80.3%	78.5%	76.8%	73.9%	71.2%

2.6 Discussion and Conclusion

We evaluated two DLNN systems, VMP-T and the VMP-L for vehicle maneuver prediction and conducted comparative studies with VMP-NN. Those systems were experimented among various prediction horizons on NGSIM dataset and UMD-ISL datasets. Our findings are as follows.

- For the remote vehicle detection, the VMP-T system gave the best detection accuracy: 84.2% on NGSIM dataset. This is achieved through the use of self-attention mechanisms that allow the model to weigh the importance of different input tokens when generating output tokens. In contrast, LSTMs and NNs have limitations in modeling long-range dependencies and may struggle to capture complex patterns in the data.

- The systems evaluated on NGSIM dataset outperformed the systems evaluated on UMD-ISL dataset. Because NGSIM dataset provides additional information about SVs, which are important for accurate vehicle maneuver prediction.
- On average, VMP-L and VMP-T outperformed VMP-NN in all domains by 17%. This is caused by that plain neural networks make predictions based on the current context only, instead of also looking backward for past information.



(a)

(b)

Figure 73. System comparison among various prediction horizons on (a) NGSIM dataset, (b) UMD-ISL.

Chapter 3: Leveraging Visual Data for Predicting Vehicle Maneuvers

3.1 Introduction

The prediction of vehicle maneuvers is a critical task for autonomous driving systems, as it enables the vehicle to anticipate the actions of other vehicles on the road and make appropriate decisions in real-time. Recent advances in computer vision have led to the development of effective algorithms for vehicle maneuver prediction that leverage visual data from cameras [33, 34, 49, 50].

The visual information in vehicle's front view image is important in predicting their maneuvers. They usually contain traffic light, traffic sign and other traffic information. For example, the turn maneuvers mostly occur at intersections, and it would be easier for the system to predict turn maneuvers accurately if it is able to recognize the driving scene.

One of the challenges in leveraging visual data for vehicle maneuver prediction is the high dimensionality and complexity of the video data. The raw video data can be noisy, and it may be challenging to identify the most relevant features for prediction. However, recent advancements in deep learning and computer vision have enabled the development of more effective algorithms for analyzing and processing visual data. Over time, neural network architectures have become deeper and more complex, with more layers and parameters [51], leading to improved performance on complex data. The increase in the number of layers in models has been driven by several factors. Second, the availability of large datasets and powerful computing resources has made it possible to train deeper and more complex networks. Finally, the development of techniques such as residual connections, batch normalization, and dropout has enabled the training of deeper networks without overfitting or vanishing gradients.

In this context, this chapter explores the use of visual data in vehicle maneuver prediction, focusing on the application of deep learning algorithms. We review the current state-of-the-art approaches in this field, including the use of convolutional neural networks (CNNs) and attention mechanisms. We also discuss the challenges and limitations of these approaches and highlight potential directions for future research.

Convolutional neural networks (CNNs) are popular in extracting visual features from images and they proved to be successful in images classification, object detection and semantic segmentation. For vehicle maneuver prediction task, CNNs can take in video data and are expected to extract the features that represent driving scene information. One of the advantages of using CNNs for vehicle maneuver prediction is that they can automatically learn relevant features from the raw sensor data. This can reduce the need for manual feature engineering and improve the accuracy of the predictions. CNNs are also capable of handling large amounts of data and can be trained on large datasets, making them well-suited for the task of vehicle maneuver prediction.

In recent years, there have been many studies that have used CNNs for vehicle maneuver prediction. Researchers have explored various architectures, including single frame[8, 10], multi-frame[40, 24, 27], and spatio-temporal CNNs[52]. These models have been used to predict a wide range of maneuvers, such as lane changes, turns, and stops. And some work [24, 53] utilized CNN model that was pre-trained for scene recognition to learn the “object-centric features” and “scene-centric features”. Research has demonstrated that object-centric features, which are features learned from an image dataset for the purpose of object recognition, can effectively aid in driving maneuver recognition and detection [8, 20]. But the negative transfer problem arises when the object features extracted for general purposes have little relevance to driving maneuvers. In [53], they investigated that scene-centric features, which is a high level abstraction learned from scene data sets, are more effective in driving maneuver recognition tasks.

Nevertheless, the images usually contain a plenty of disturbing information, such as trees, sky and buildings, we want the system to concentrate on maneuver-related information as much as possible. Attention mechanisms could be a solution to this problem, because attention mechanisms allow models to focus on the most relevant parts of the input data, leading to more accurate predictions and better generalization performance. Recent researches have applied attention models in detecting driver intentions and behaviors [25, 26, 19]. Attention is a concept derived from human vision system where attention mechanism plays an essential role. It could be represented in a heat map generated from a front view video frame. A region in the heat map contains higher values if the corresponding region in the video frame is more important than other regions during driving maneuvers. As shown in Fig. 14, these areas contain the information useful

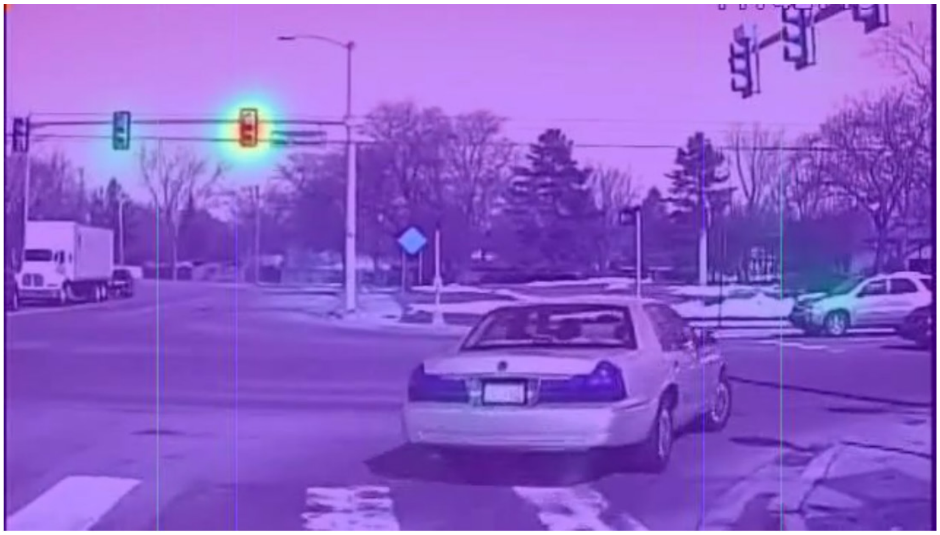


Figure 14. A heat map generated from a video frame while target vehicle was making a right-turn maneuver. Most vehicles make turn maneuver at intersections and the highlighted traffic light a crucial clue for system to determine if it’s an intersection scene.

in determining if the current scene related to vehicle maneuvers. Attention models can be generated using deep learning techniques and large amounts of annotated video sequences of driving trips. Each frame was annotated with the regions where driver's gaze was registered. This inspired us to incorporate driver attention information into our system for predicting driving maneuvers.

Researchers have explored various types of attention mechanisms, including soft attention and hard attention[54, 55]. These mechanisms have been integrated into various deep learning models, including recurrent neural networks, convolutional neural networks, and transformers.

Soft attention [55] is a commonly used attention mechanism that allows the model to selectively focus on different parts of the input data, such as the location and movement of other vehicles on the road. Soft attention assigns different weights to different parts of the input data, which can be adjusted during training to focus on the most relevant information for predicting the vehicle's future movements.

Soft attention works by calculating a weight vector for each feature in the input data. The weight vector is calculated based on a learned function that takes as input the current state of the model and the feature being considered. The weight vector is then multiplied by the feature vector to produce a weighted feature vector, which is passed through the rest of the network.

Hard attention[54] is an attention mechanism that allows the model to selectively choose which parts of the input data to use in making predictions. Hard attention can be useful when there is a large amount of noisy or irrelevant data that may negatively impact the performance of the model.

Hard attention works by making a binary decision for each feature in the input data, selecting some and ignoring others. The decision is based on a learned threshold value, which is determined during training by minimizing the loss function. Features whose values exceed the threshold are selected and used in making the prediction, while those whose values are below the threshold are ignored.

Besides the front view video, it would also be helpful if the VMP system can access to driver’s view video that contains driver’s behavior during driving. Driver's view video provides a first-person view of the environment from the driver's perspective, and can be used to provide important contextual information that may not be captured by other types of sensors, such as radar or LiDAR. Driver’s behavior is usually highly related to the decision making and reasoning processes of vehicle maneuvers. For example, a driver glances at mirrors with a 65%−92% probability before making a lane change [56]. The use of driver's view video in vehicle maneuver prediction is an emerging area of research that has shown great potential in improving the performance and reliability of autonomous driving systems [37, 38]. These studies inspired us to utilize not only front view video, but also the driver’s view video, which recorded the driver’s facial expression and upper body motion.

In this chapter, We demonstrate the effectiveness of fusing driver attention heat map, the driver’s head movement and vehicle signals for the prediction of driving maneuvers in complicated dynamic traffic environment.

The rest of the chapter is organized as follows. Section 3.2 reviews the literatures about computer vision based VMP systems. In Section 3.3, we describe the attention models that are designed to learn the visual information in front view videos, and the algorithm for driver face shift feature extraction. We present the design and the results of empirical experiments in Section 3.4 and Section 3.5, and then conclude the chapter in Section 3.6.

3.2 Related Work

3.2.1 Attention Mechanisms Related to VMP systems

Deep neural networks have been developed to learn the Regions Of Interest (ROI), i.e., the regions in which the human drivers are likely to pay attention [25, 26, 27, 28], such as regions containing traffic lights, traffic signs, vehicles and other road users. Andrea et al. [25] introduced a computer vision approach to model the human attentional behavior during driving environments. They developed a deep learning technique for predicting where a driver would be looking at in a specific driving situation. The model was trained on DR(eye)VE dataset [29] and it has three input data streams, a) a RGB video frame, b) motion cues (in terms of optical flow) and c) semantic segmentation. The output were the estimated attentional maps. Ye et al [26] proposed a different deep network model, FCN-ConvLSTM, designed to learn the areas of scenes where attention of human drivers should be placed. The FCN-ConvLSTM was trained on the Berkeley Deep Drive Attention dataset (BDDA), which contains 1232 videos annotated by following drivers' eye movements while they were watching recorded driving videos.

In [59], the multi-head attention is used to model the vehicle interaction and encode lane features. The model takes the trajectories of TV and SVs as inputs to output the distribution of the future trajectories. The prediction model has an encoder-decoder structure that uses multi-head attention. The encoder maps past trajectories and lane information to a compressed representation, which is then used by the decoder to generate predictive mean and predictive covariance for future trajectories. The model has two attention layers: a vehicle attention layer and a lane attention layer. Both layers use the scaled dot product attention function to compare queries, keys, and values. The encoder has both vehicle attention and lane attention layers, which generate attention-based representations for each surrounding vehicle. The decoder consists of a single vehicle attention layer that gathers the encoded information to predict the trajectories of surrounding vehicles. The outputs of the decoder are predicted mean and predicted covariance, which are modeled as a Gaussian distribution. Overall, the model is designed to use the relationships among the past trajectories of the vehicles and the lane information to make accurate predictions about future vehicle trajectories. The model was evaluated on real-world trajectories in NGSIM dataset and has an error value of 0.89m in the longitudinal direction and 0.11m in the lateral direction over a 3-seconds prediction horizon.

Furthermore, the attention model in [58] was used to models the interaction between different traffic participants. They proposed an attention based vehicle motion prediction system that consists of three main components: an Encoding Layer, a multi-head Attention module, and a Decoding Layer. The Encoding Layer uses an LSTM encoder to capture the historical information of the vehicle's motion and encode it in a useful format for prediction. The Attention module links the hidden states of the encoder and decoder and extracts the importance of the surrounding vehicles based on their spatio-temporal encoding. It then uses different operations to determine the future motion of the target vehicle and form a vector representing the context influence. Finally, the Decoding Layer receives the context vector and generates parameters of the distribution over the target vehicle's predicted future positions. The combination of these three components allows the model to consider the relationships and interactions that occur on the road to make more realistic predictions about vehicle motions. The model takes into account the historical and contextual information of vehicles, which can lead to more accurate predictions of vehicle motions. The proposed model can be used in a variety of applications, such as autonomous driving and traffic flow analysis, to improve safety and efficiency on the road. Their proposed model is evaluated on NGSIM dataset and achieved 3.83 meters of RMSE over a 5-second prediction horizon, which outperforms the state-of-the-art performances.

Based on the advances in [58, 59], paper [60] used multi-head attention to model the interactions between TV and the combined context features. The model described in [118] is designed to predict future vehicle trajectories using a combination of past trajectory and map data. A social tensor is generated by placing trajectory encoder states of surrounding agents in a 2D spatial grid and concatenated with map features to create a combined representation of agent motions and the map. Multi-head attention is used to extract salient parts of the joint representation, with an attention head assigned to each mixture component. The output of each attention head is calculated as a weighted sum of value vectors, and each output is concatenated with the target vehicle trajectory encoder state to create a context representation for each mixture component. Each context vector is then fed to an LSTM decoder to generate predicted parameters of the distributions over the target vehicle's estimated positions for the next time steps. The model was evaluated on a public NuScenes dataset and achieved state of the art results on the NuScenes prediction benchmark.

When we expand the research field from the maneuver prediction for vehicle only to for all traffic participants, we found there are more attention based models have been studied. Paper [57] proposes an attention version of the social LSTM model, which is a popular deep learning method for predicting pedestrian trajectories. The original social LSTM model represents pedestrians in the local neighborhood using LSTMs and generates their future trajectory by systematically pooling relevant information. Paper [57] improves upon this architecture by introducing a soft and hardwired attention framework to more efficiently embed the local neighborhood information. The authors demonstrate the importance of fully capturing contextual information, including the short-term history of the pedestrian of interest as well as their neighbors. By using attention mechanisms to focus on the most relevant parts of the input sequence, the proposed approach is able to achieve better performance on pedestrian trajectory prediction tasks.

3.2.2 The Use of Driver's View Videos in VMP System

Researchers have demonstrated that facial features could be used in driving maneuvering detection. Authors in [40] developed a facial landmark point detector for tracking landmark points on driver's face, and then optical flow trajectories were generated from these fixed points. Furthermore, they projected 2D landmark points to 3D head model to estimate three head poses of yaw, pitch and row. These new features were combined with vehicle signals for training a maneuver prediction model. They showed that the system used these new features performed 6% higher in precision better than the model without these features.

Face landmark detection is a research area within computer vision that focuses on detecting key points or landmarks on a face, such as the corners of the eyes, the tip of the nose, or the corners of the mouth. Accurate detection of face landmarks is important for a wide range of applications, including facial recognition, emotion recognition, and virtual reality. In recent years, deep learning models have shown significant improvements in the accuracy and efficiency of face landmark detection. These models typically use convolutional neural networks (CNNs) to extract features from the input image, and then use fully connected layers to predict the location of the landmarks.

One of the most popular datasets used for training and evaluating face landmark detection models is the 300-W dataset, which consists of over 600 images with annotated facial landmarks. Researchers have developed a variety of deep learning models for face landmark detection, including fully connected neural networks, convolutional neural networks, and recurrent neural networks. In addition to improving the accuracy of face landmark detection, researchers have also explored ways to make these models more robust to variations in lighting, pose, and facial expression. One approach is to use data augmentation techniques to generate additional training examples with different lighting conditions, poses, and expressions. Another approach is to use adversarial training, where the model is trained to generate images that are more difficult to detect landmarks on, in order to improve its robustness.

Paper [38] shows a state-of-the-art face landmark detection model, which is Convolutional Experts Constrained Local Model (CE-CLM). It combines the strengths of the Constrained Local Model (CLM) and Convolutional Neural Networks (CNNs). The CLM is a traditional model for face landmark detection that uses a shape model to constrain the search space for facial landmarks. However, it can struggle to handle variations in pose and lighting. The CE-CLM model addresses this issue by using CNNs to learn local appearance models for each landmark, allowing it to handle more complex variations. The CE-CLM model consists of a CNN feature extractor that processes the input image and generates feature maps, which are then used to train a set of experts, each responsible for predicting the location of a specific landmark. During inference, the experts are combined using a set of weights that are learned through a constrained optimization process, ensuring that the landmarks are consistent with the shape model. Compared to traditional CLM models, CE-CLM has been shown to improve the accuracy and robustness of face landmark detection, especially in challenging conditions such as varying illumination and occlusion. It has been used in a variety of applications, including facial expression analysis, facial recognition, and virtual reality.

In this research we explore the use of drivers' facial features for predicting vehicle's maneuvering actions.

3.3 Visual Feature Extraction for VMP systems

3.3.1 Attention Feature Extraction

a. Convolutional neural networks

Since our attention model is based on a convolutional neural network, so we introduce how it works on images first. Convolutional neural networks (CNNs) are a type of artificial neural network used in deep learning for processing input data that has a grid-like structure, such as images, videos, and speech signals. A CNN consists of several layers, each designed to extract different features from the input data. The first layer is usually a convolutional layer, which applies a set of kernels to the input data and produces a set of feature maps. In the case of images with multiple channels (e.g. RGB), the kernel has the same depth as that of the input image. To obtain a convoluted feature output with one-depth channel, the kernel is multiplied with each channel of the image using matrix multiplication as shown in Fig. 15. The resulting products are then summed together with the bias term, resulting in a squashed output. These filters are learned during training, allowing the network to automatically learn features such as edges, corners, and textures.

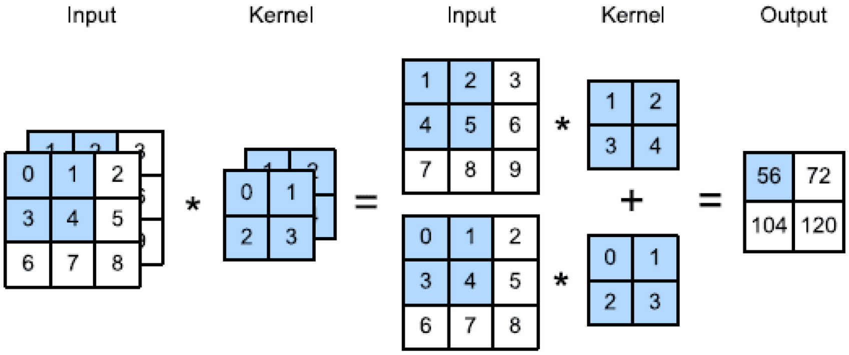


Figure 15. Convolutional operations on input matrix.

The output of the convolutional layer is typically passed through a non-linear activation function, such as ReLU (Rectified Linear Unit), which introduces non-linearity into the model and allows it to capture more complex patterns in the data. The ReLU function is defined as:

$$\mathrm{f(x)} = \max(0, x)$$

where x is the input to the function. The output of the function is simply the maximum of the input and 0. This means that if the input is positive, the output is equal to the input, and if the input is negative, the output is equal to 0. Leaky ReLU is a type of activation function used in artificial neural networks, and is a variant of the Rectified Linear Unit (ReLU) activation function. While ReLU sets negative inputs to zero, Leaky ReLU introduces a small positive slope to negative inputs, preventing the dying ReLU problem. The Leaky ReLU function is defined as:

$$\mathrm{f(x)} = \max(0.01x, x)$$

where x is the input to the function. When x is positive, the output is simply x, as in the ReLU function. However, when x is negative, the output is a small fraction of x (typically 0.01) multiplied by x, which introduces a small positive slope for negative inputs. Leaky ReLU has several advantages over ReLU, including better performance in certain types of deep learning models, and reduced likelihood of neurons becoming inactive.

After the activation function, the feature maps are usually down sampled or pooled, which reduces the dimensionality of the feature maps and increases the computational efficiency of the model. They reduce the computational power needed to process data, while also extracting dominant features that are invariant to rotation and position. This helps ensure that the model can be effectively trained. Pooling operators use a fixed-shape window that is moved over the input data according to its stride, producing a single output for each location the window passes over. This is similar to how convolutional layers work, but with no parameters or kernel involved. Instead, pooling operators are deterministic and calculate either the maximum or average value of the elements in the window, known respectively as max pooling and average pooling.

During max pooling, a window (usually 2x2 or 3x3) is moved over the input feature map with a fixed stride as shown in Fig 16. For each position of the window, the maximum value within the window is taken as the output value. The resulting output feature map has reduced spatial resolution and fewer features. It also can help introduce translation invariance by selecting the most prominent features regardless of their exact location within the window

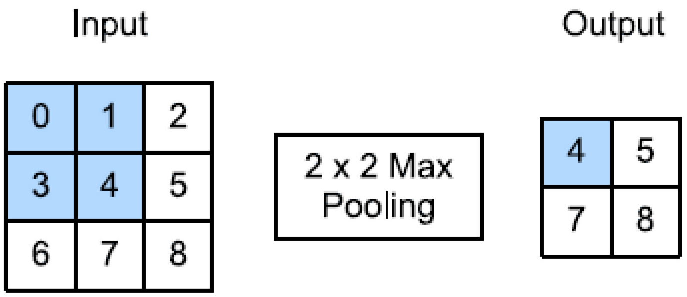


Figure 16. Max pooling operation.

During average pooling, a window (usually 2x2 or 3x3) is moved over the input feature map with a fixed stride. For each position of the window, the average value within the window is taken as the output value. The resulting output feature map has reduced spatial resolution and fewer features.

Average pooling is similar to max pooling, but instead of selecting the maximum value, it takes the average value within the window. This can be useful in situations where max pooling may be too aggressive and result in loss of important information. Average pooling can help smooth out the features in the input feature map and provide a more generalized representation of the input.

However, average pooling has some limitations. It can lead to information loss and reduce the ability of the network to capture fine-grained details in the input. Additionally, it may not be as effective at introducing translation invariance as max pooling.

b. Our proposed attention model

Since drivers obtain the most traffic information through views of surrounding scenes, in particular the front road view, drivers attentions related to maneuvering can be learned through the front view videos. However, it is difficult to learn a complex model with limited training data. In this research, we use the following procedure to generate attention features in video images that contained events of driving maneuvering.

First we used the VGG19 model [9] to generate useful visual features from traffic scenes. The VGG19 model was pre-trained on Palaces-365 [61] for the image classification. The features generated by VGG19 contain semantic information of traffic scene and are important for driving maneuver detection. For VGG19 model, the input is a 224*224*3 image and the output is a L dimensional feature vector $F_t = \{f_t^1,\ f_t^2, \dots ,\ f_t^L\}$. Although the images of driving scenes provide rich clues of driving maneuver, they also include irrelevant information such as roadside trees, sky and buildings. To solve this problem we propose the following attention model to detect the region that is important for the driver to pay attention.

According to Chapter 2, we proposed VMP systems based on LSTM network and Transformer network. I will introduce how to integrate them with attention model respectively.

The attention model takes feature F_t and LSTM hidden state h_{t-1}at previous time as input to generate an attention heat map. The Attention heat map A_t is a vector of the same length L as with feature F_t and it's represented by $A_t = \{a_t^1,\ a_t^2, \dots ,\ a_t^L\}$, where the element a_t^i is the probability of the region being focused on by the driver during driving. The sum of the attention values on all regions of input image should be 1, $\sum_{i=1}^{L} a_t^i = 1$. The attention weight value a_t^i is generated from alignment score e_t^i through softmax function:

$$a_t^i = \frac{\exp(e_t^i)}{\sum_{i=1}^{L}\exp(e_t^i)} \ . \qquad (4.1)$$

The alignment score e_t^i measures how well the input feature f_t^i matching with the previous decoder output h_{t-1} using the following formula:

$$e_t^i = W_{att}\tanh(f_t^i, h_{t-1}) \qquad (4.2)$$

where W_{att} is the weight matrix that can be learned in attention model.

Based on features vector $F_t = \{f_t^1,\ f_t^2, \dots ,\ f_t^L\}$ and attention heat maps $A_t = \{a_t^1,\ a_t^2, \dots ,\ a_t^L\}$, the attention features are obtained by multiplying F_t with the attention probability values, $v_t = g_{flatten}\ (a_t \times f_t\ )$.

For Transformer based VMP system, as we discussed in Chapter 2.3.3, Transformer model takes data points in a sequence simultaneously and utilizes self-attention mechanism on them. We don't need to calculate attention weights separately. So the visual feature F_t will be sent to VMP-T directly.

3.3.2 Driver Face Shift Feature Extraction

The face shift features are calculated based on the face landmarks extracted by the CLM (Constrained Local Model) presented in [37, 38]. The CLM detects and tracks 68 fixed landmark points on the driver's face image in a video sequence. These landmark points were labeled to localize and represent regions of the face, such as eyes, eyebrows, nose, mouth and jawline. The CLM was trained on the 890 frontal face images from CMU Multi-PIE dataset [62]. The training data contains various head poses and illumination conditions, which are similar to the imaging environment of our dataset (ISL-UMD dataset). We use the CLM to extract the 68 landmark points in each image in every video sequence.

The face shift features in DMD at time t are estimated by tracking the locations of the 68 landmark points in the image frame at the current time t and the previous frame t-1 using the formula below

$$M_t = \frac{\sum_{i=1}^{68} L_i^t - L_i^{t-1}}{68} = \{(\frac{\sum_{i=1}^{68} x_i^t - x_i^{t-1}}{68}, \frac{\sum_{i=1}^{68} y_i^t - y_i^{t-1}}{68})\} \quad (3)$$

where $L_i^t = \{(x_1^t, y_1^t), (x_2^t, y_2^t), \dots, (x_{68}^t, y_{68}^t)\}(x_i^t, y_i^t)$ are the coordinate of the i_{th} landmark at time t and $M_t \in R^2$.

Fig. 17 shows an example of a left-turn maneuver video sequence from our dataset. The top frames are selected from the event video sequence, the frame 50 to 51 show the driver was looking ahead and the corresponding face shift curves didn't change before the event. The frame 207 and 208 show the driver was looking around to check whether the traffic situation was safe to make a turn. The frame 463 and 464 were recorded after the maneuver event, in which the driver was looking ahead without shifting. We showed the corresponding horizontal and vertical face shift

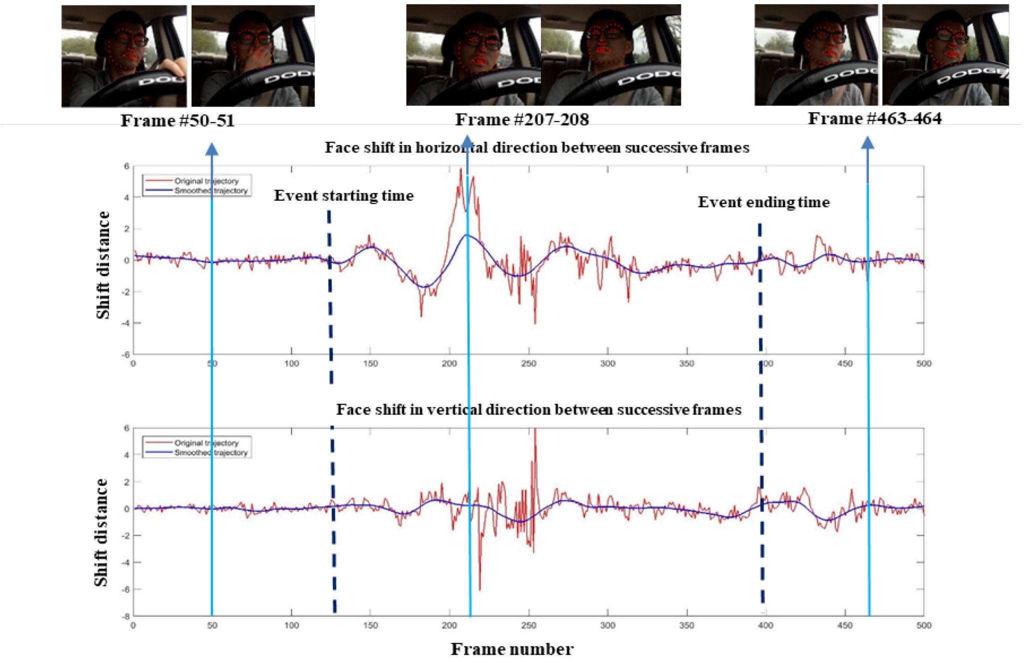


Figure 17. A left-turn maneuver, the driver glances at the mirror before making a left turn, which shows high relevance between face shift (horizontal and vertical) and driving maneuver.

displacements below the video frames. The horizontal displacement curve shows high fluctuation at the same time as the driver was moving his face. This example demonstrates the correlation between driver face shifts and driving maneuvers. Since in most times drivers turn their heads to look at left and/or right before and during a driving maneuver, the face shift in horizontal direction has more significant features than the vertical direction with respect to the driver's maneuvers.

3.4 Experiment Setup

We present empirical experiments using real world datasets to evaluate the performance of VMP systems that learns visual information.

3.4.1 Dataset and Evaluation metrics

In the experiments in chapter 3, we continue to use UMD-ISL dataset since it contains both of front view video and driver's view video. The data can be represented by:

$$\mathbf{Z}_t = [\, \mathbf{z}_{t-(w-1)} | \dots \;|\; \mathbf{z}_{t-2} \;|\; \mathbf{z}_{t-1} \,|\, \mathbf{z}_t \,]$$

where $\mathbf{z}_t = [v_t, M_t, x_t]$ for LSTM based VMP system, v_t denotes the attention feature extracted from the front view video, M_t denotes driver's face shift feature extracted from the driver's view video, and x_t denotes the vehicles signals that contain GPS, speed and headings of EV. For Transformer based VMP system, $\mathbf{z}_t = [F_t, M_t, x_t]$ and F_t denotes the visual feature extracted by Inception V3.

The systems in this chapter will also be measured by using F1 score and showed in Fig 18.

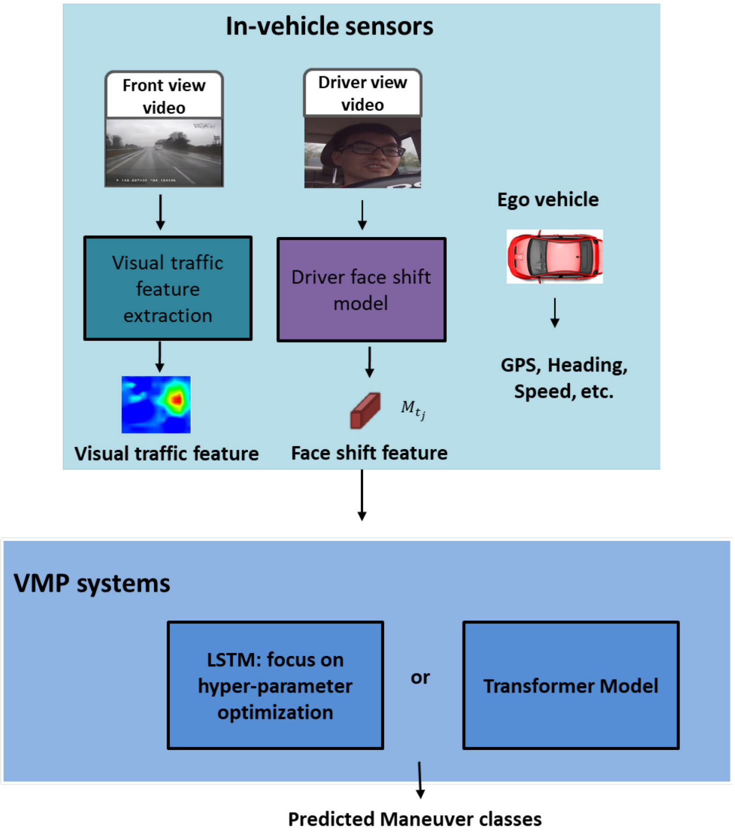


Figure 18. Ego vehicle maneuver prediction using in-vehicle sensors' data in UMD-ISL dataset.

3.4.2 Baseline models

To illustrate the effectiveness of the proposed visual features in VMP system, we compared following 4 systems,

- VMP-L: The system we proposed in chapter 2, visual features are not used.
- VMP-T: The system we proposed in chapter 2, visual features are not used.
- VMP-LV: Based on VMP-L, visual features are concatenated with vehicle signals as input data.

- VMP-TV: Based on VMP-T, visual features are concatenated with vehicle signals as input data.

3.4.3 Hyper-parameter Settings

In the experiments in this chapter, we used the following hyper-parameter settings:

- Data sampling rate: 10 Hz
- Window size: $w = 5$ seconds
- Number of output classes: 5
- Max number of training epochs: 500
- Number of LSTM layers: 1,2,3
- Number of heads in VMP-T: 6
- Number of encoder blocks in VMP-T: 3
- Look ahead prediction time: $\Delta t = 1, 2, 3, 4, 5$ seconds
- Learning rate: 0.001
- Batch size = 64

In our experiments, UMD-ISL data were split into 70% training set and 30% validation set. With the above hyper-parameters, the systems using video data have 142,277 and 85,930 weights. The training time for a VMP-T model was about 17 hours, while a VMP-L model took about 12 hours.

3.5 Experiment Results

The experiments in the table show the F1 score of four different vehicle maneuver prediction systems, VMP-L, VMP-T, VMP-LV, and VMP-TV, at different prediction horizons. VMP-L and

VMP-T use LSTM and transformer models, respectively, while VMP-LV and VMP-TV extend VMP-L and VMP-T by adding visual features as input data.

The results indicate that incorporating visual features into the prediction model, as done in VMP-LV and VMP-TV, leads to significant improvements in prediction accuracy. Specifically, both VMP-LV and VMP-TV outperform VMP-L and VMP-T, respectively, across all prediction horizons. Moreover, VMP-TV achieves the highest F1 score for all prediction horizons, suggesting that the transformer model with visual features as input is the most effective system for vehicle maneuver prediction.

Table 3.1: F1 score for vehicle maneuver prediction on UMD-ISL dataset

System	Prediction Horizon Δt (s)				
	1	2	3	4	5
VMP-L	75.1%	72.9%	69.4%	67.5%	64.0%
VMP-T	80.3%	78.5%	76.8%	73.9%	71.2%
VMP-LV	86.3%	84.6%	81.2%	75.1%	74.5%
VMP-TV	88.6%	86.9%	85.5%	83.7%	80.2%

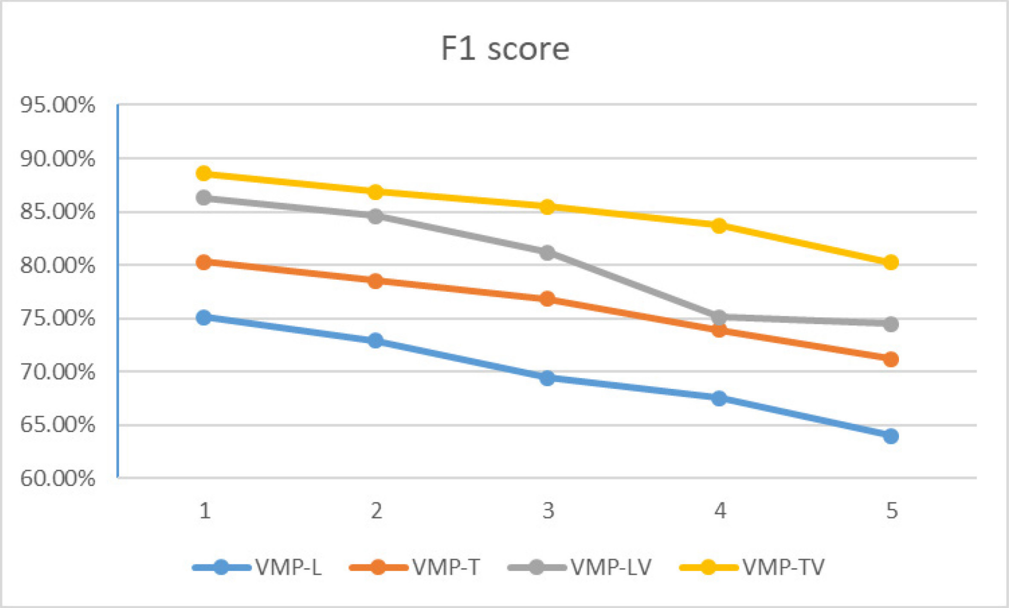


Figure 19. System comparison among various prediction horizons on UMD-ISL dataset.

3.6 Discussion and Conclusion

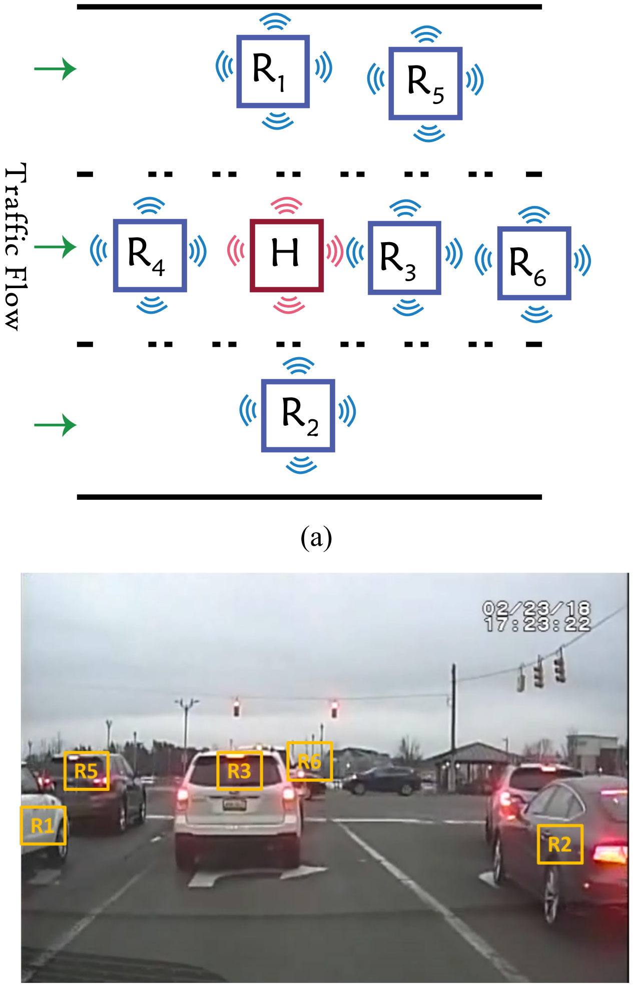
In conclusion, the results demonstrate the importance of incorporating visual features into vehicle maneuver prediction systems. While VMP-L and VMP-T are effective, VMP-LV and VMP-TV achieve superior performance, with VMP-TV being the most accurate system. These findings suggest that future research in vehicle maneuver prediction should consider the incorporation of visual features as input to improve prediction accuracy.

Chapter 4: Target Vehicle Position Prediction using V2V Communication and Deep Learning Neural Networks

4.1 Introduction

The capacity to evaluate and comprehend the context of a traffic situation in real-time, also known as situational awareness, is crucial for human drivers as well as advanced driver assistance systems (ADAS). In this study, we characterize context and situational awareness as the data a vehicle requires to execute secure maneuvers within a typical traffic setting, which includes other vehicles performing various actions like changing lanes, making turns, overtaking, yielding, and so on. A critical aspect of attaining this situational awareness involves a vehicle's ability to recognize and anticipate the relative locations of nearby vehicles. Many advanced driver assistance systems (ADAS) have been developed to help drivers handle various traffic scenarios safely [63].

This chapter focuses on the situational awareness illustrated in Figure 19. In Figure 19(a), the host vehicle H is driving on a road with three lanes, in proximity to surrounding vehicles, termed remote vehicles and labeled as R1, R2, …, R6. Note that those remote vehicles were numbered and labeled randomly. To analyze the traffic circumstances surrounding the central vehicle H, it is necessary to acknowledge that distant vehicles like R1 and R2 are traveling parallel in both left and right adjacent lanes. Also, vehicles such as R3 and R4 are in the same lane, positioned both ahead and behind, with a vehicle like R5 ahead in the left adjacent lane. An interesting aspect of this situation is the occurrence of R6, which may be entirely concealed from vehicle H's perspective; for example, consider R3 being a sizable SUV while R6 is a motorbike or a small car. However, R6 still presents a potential risk to vehicle H, as if R6 were to suddenly apply the brakes,



(a)

(b)

Figure 20. (a) A host vehicle **H** and multiple remote vehicles, **R1** ~ **R6** are in communication. (b) Remote vehicles seen by the host vehicle.

the combined reaction time delays of R3 and the host could make it difficult for vehicle H to prevent a collision involving three vehicles. This particular situation will be referred to as the H - R3- R6 scenario.

Many researchers have worked on improving radar and camera-based ADAS systems for preventing crashes and understanding traffic situations for ADAS and autonomous vehicles [64] – [69]. However, these technologies that are installed in vehicles have some drawbacks, such as a limited range and a need for a clear line of sight. For instance, camera-based systems may not work well in the H - R3- R6 scenario mentioned above, where human drivers would also struggle.

Moreover, camera systems may perform poorly when the weather is bad, the lighting changes, or there are obstructions.

Vehicle-to-vehicle (V2V) communications have attracted a lot of attention lately. In this approach, vehicles share real-time information about their position, speed, and direction. This information can help determine where other vehicles are on the road and alert drivers about possible dangers. Several V2V communication technologies and protocols can be used in commercial vehicles, such as DSRC and cellular technologies like 4G or 5G mobile systems. V2V communications have some benefits over sensor systems that are installed in vehicles, such as a longer range and no need for a clear line of sight [70] – [71]. This means that V2V based ADAS can predict some potential danger traffic situations and warn drivers accordingly earlier than the systems based on sensors, cameras, or radar can. The National Highway Traffic Safety Administration estimates that V2I and V2V applications can prevent 80% of crashes that are not caused by impairment [72]. Although V2V communication technologies have some disadvantages like signal loss, security and privacy concerns and cannot deal with dangerous situations such as road and lane departure [70], many studies agree on that reliable ADAS should include a V2V communication component to complement the sensor systems that are installed in vehicles to improve the performance of the overall vehicle safety system in terms of warning timing, false alarm reduction, and crash scenario coverage [71].

This chapter describes our work on developing advanced deep learning neural networks for accurate detection and prediction of the relative position of a remote vehicle to a host vehicle. The problem is formulated as a classification problem where the classes represent the situational awareness, i.e. different possible positions that the remote vehicle can have relative to the host one; for instance in front in the left neighboring lane (class 1), behind in the same lane (class 7), and so on. This research builds on our work presented in [72], where we described a machine learning framework, which uses a feature vector extracted from the V2V data communicated between the host and remote vehicles. An intelligent system, which we referred to as Geo+MLP, was developed using geometric modeling and a Multi-Layer Perceptron (MLP) neural network. We evaluated the Geo+MLP system using the V2V data obtained from 41 actual driving trips captured by a fleet of more than 1800 vehicles equipped with DSRC communication devices. In this paper, we focus on developing highly accurate intelligent systems for predicting the relative position of a remote vehicle to a host vehicle in long horizons: 0.1 ~ 3 seconds. We propose two types of deep learning systems: LSTM and transformer-based systems, as well as three sets of features extracted from V2V communication signals. Experimental results generated from 69 trips show that both the LSTM and transformer-based systems outperformed MLP networks in remote vehicle position detection by 1.8% ~ 4.7%, and prediction by 10.6% ~ 17.8%. In particular, when the prediction is made for longer horizons, e.g., prediction for 1, 2 and 3 seconds ahead, the LSTM and transformer systems outperformed MLP by 14% ~ 25%.

The remainder of this chapter is organized as follows. Section 4.2 reviews the related work in the area of relative position detection and prediction. In Section 4.3, we introduce the effective features extracted from V2V signals for remote vehicle position detection and prediction. Section 4.4 introduces the two DLNN models: LSTM and Transformer, with applications to remote vehicle position detection and prediction. Section 4.5 and 4.6 present experimental settings and results using naturalistic V2V driving data collected in a U.S. city. Finally, Section 4.7 provides a summary and conclusion of the research results.

4.2 Related Work

Vehicle position detection and predictions are important technologies for building situational awareness, such as pre-crash detection, prediction, and avoidance. Most of the research in this area has used sensors on the vehicle, such as radar, cameras and/or LIDAR devices, along with machine learning algorithms for solving related problems for ADAS, such as detecting vehicles, pedestrians, traffic signs, etc. Some of these technologies for safe driving have been implemented in market, such as systems for warning about forward collisions and systems for detecting blind spots. However, these systems based on sensors on the vehicle are not dependable in various scenarios, such as dim lighting, poor weather and occlusions—conditions where the driver would need an assistance system the most. V2V communication has the capabilities of providing traffic information to vehicle safety systems under these conditions [71]. In addition, V2V communication such as DSRC has a longer capability range (300 meters) than ultrasonic sensors, cameras, and radars, and can therefore alert drivers of dangerous situations earlier and more effectively [70]. Moreover, V2V can be combined with radars and cameras to achieve even greater safety [66]. For example, V2V can alert drivers to blind spots, increase awareness during lane changes, and when passing a vehicle on a two-lane road requires crossing into oncoming traffic [65]. V2V is also useful in emergency braking situations where the tail (emergency brake) lights on cars are obscured, such as at intersections and left turns, which have the highest crash incidence [73].

One important aspect of a vehicular safety application that uses V2V communication is to determine a way to accurately predict the position of vehicles and counteract the inherent position bias and random errors in low-cost GPS devices commonly used in the car industry [6]. A number of vehicle relative positioning techniques or vehicle trajectories have been developed solely using

V2V data, or using both GPS data and video data to detect the relative positions of a remote vehicle with respect to the host vehicle [4-7]. In [66], the authors used a simple geometric method to calculate the appropriate relative angles of remote vehicles based on GPS coordinates of the host and the remote vehicles and an image analysis method to detect whether a vehicle is traveling in the same direction and in the same lane. The same lane and same direction test was done by taking the picture of the license plate of the vehicle ahead of the host vehicle, and comparing the license plate number in the image with the number extracted from the message in VANETS. By conducting the experiments using the GPS and image data obtained from a smart phone, the authors found that the image processing approach was more accurate in determining remote vehicle positions than the GPS-based method, but it also required much more computational power.

Kalman filter-based systems have been popularly used to predict the positions of remote vehicles. However, as the authors in [67] point out, basic prediction and estimation techniques that use Kalman filters are effective on straight roads. However, when it comes to curved roads, these methods can produce inaccurate predictions that may even fall outside the road's boundaries. To enhance the precision of predictions on curved roads, the authors in reference [67] introduced a system that employs 4 different Kalman filter models. Every model is effective for a particular set of circumstances. These models offer a mathematical formula that can be utilized to forecast the future position of the vehicle. The proposed system demonstrated successful performance in experimental trials. However, its prediction time was 3 seconds - twice as long as the average human reaction time of 1.5 seconds.

With the success of recurrent neural networks (RNN), in particular, LSTM networks, in modeling non-linear temporal dependencies in sequence learning and generation tasks, recent research has focused on using these neural networks in driver maneuvering prediction [53, 74, 10], and vehicle trajectory prediction [75, 76, 43, 44, 39].

4.3 Extracting Effective Features For Remote Vehicle Prediction

Fig. 20 shows an overview of the proposed deep learning-based system for detecting and predicting remote vehicle positions based on V2V communication data. In the first stage of processing, the raw data obtained from the V2V communication between the host and remote vehicles can be denoted as: $\mathbf{V2V}_H(t)$ and $\mathbf{V2V}_R(t)$ are used to generate a geometric feature vector $\mathbf{x}(t)$ that contains the information of the relative position of the remote vehicle, which are discussed below. The second stage is a deep learning neural network that detects and predicts the relative position of the remote vehicle based on $\mathbf{x}(t)$. Two different deep neural networks are investigated: an LSTM-based network and a transformer model-based neural network. The LSTM systems are gated recurrent neural networks, and the transformer model does not use recurrence but relies entirely on an attention mechanism to draw global dependencies between input and output. Both systems have been discussed in Chapter 2.

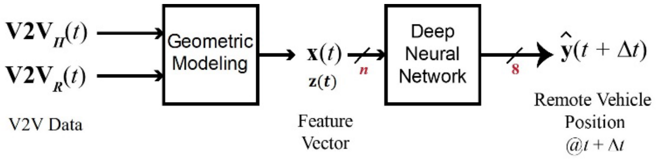


Figure 21. Overview of the proposed intelligent system for remote vehicle position detection and prediction.

The relative vehicle position detection and prediction can be defined as a classification problem with 8 classes, as illustrated in Fig. 21. Here, the host vehicle is surrounded by remote vehicles and their possible positions are given. For example, class 7 indicates the remote vehicle

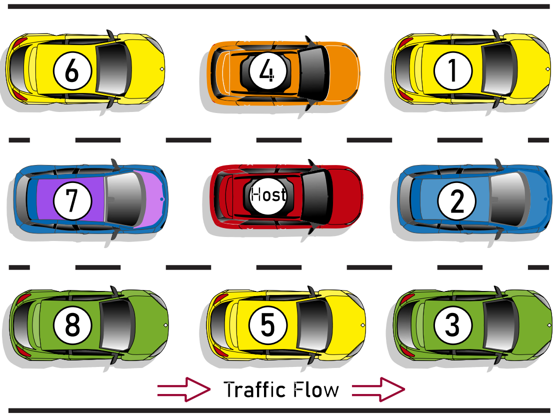


Figure 22. Annotation of eight position classes for the remote vehicle relative to the host.

is behind of the remote vehicle in the same lane, and class 3 indicates that the remote vehicle is ahead of the host vehicle in the right adjacent lane, etc. This type of information provides important situational awareness to the host vehicle that can be used in the development of both ADAS systems and autonomous vehicle systems.

We assume the raw V2V communication data shared between the host and remote vehicles: $\mathbf{V2V}_H(t)$ and $\mathbf{V2V}_R(t)$, contains the following information:

1. Time stamp (s)
2. GPS Latitude
3. GPS Longitude
4. Vehicle speed (m/s)
5. Heading (radians)

As described in [72], the latitude and longitude (in degrees) are transformed to Euclidean coordinates using a UTM transformation. Next, the coordinates are normalized so that the host is at the origin traveling due east, as shown in Fig. 22. Here, $\mathbf{H}(t)$ denotes the position of the host at $t = 0$ (which is the origin after normalization), $\mathbf{H}(t-1)$ is the position of the host at time $t-1$.

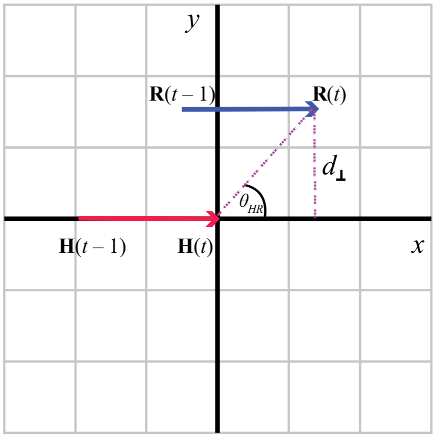


Figure 23. Normalized position of both the host and remote vehicles.

Similarly, $\mathbf{R}(t)$ and $\mathbf{R}(t-1)$ represent the position of the remote vehicle at times t and $t-1$, respectively. After normalization, the host is heading due east; the angle θ_{HR} between the host and remote vehicle is computed, as well as the perpendicular distance $d_\perp$.

In our previous research [72], we derived three geometric features that characterize the relative position of remote vehicle relative to the host vehicle: $\theta_{HR}(t), d,$ and $d_\perp$, where d is the Euclidean distance, $d_\perp$is the perpendicular distance, and $\theta_{HR}(t)$ is the relative angle between the host and remote vehicle at time *t*:

$$\theta_{HR}(t) = \text{atan2}(\frac{y_R(t)}{x_R(t)})$$

where atan2 is the arctangent function that outputs the full range of angles: $[-\pi, \pi]$.

The measures $d_\perp$ and θ_{HR} can be used to determine the relative position of the remote vehicle with respect to the host. In particular, by using a suitable threshold $T_{d_\perp}$, the host and the remote vehicles are in the same lane when $d_\perp \leq T_{d_\perp}$and are in adjacent lanes otherwise. Similarly, using a set of angular thresholds, as shown in Figure 23, θ_{HR} is used to determine which of the 8 possible output classes the remote vehicle is in (with respect to the host).

The 3 primary features described above can be augmented with various V2V data. In this study, we examined 3 different sets of features to represent the host and remote vehicles:

- 3F: three primary geometric features that were introduced above: $d(t)$, $d_{\perp}(t)$, and $\theta_{HR}(t)$.
- 9F: : the 2 dimensional coordinate vectors $\mathbf{H}(t-1)$, $\mathbf{R}(t)$, and $\mathbf{R}(t-1)$ are concatenated with three primary features together. Since $\boldsymbol{H}(t) = 0$, it is not included here.
- 11F: incorporates the features used in the 9-Features model and adds the speed of the host and remote vehicles at time *t*: $v_H(t)$ and $v_R(t)$, respectvely.

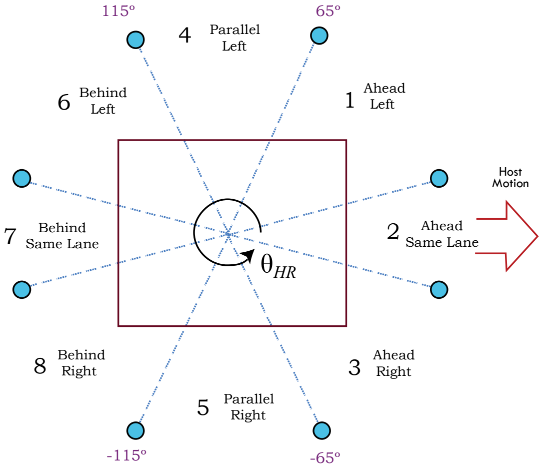


Figure 24. The relative angle θ_{HR} provides information of remote vehicle's relative position.

4.4 Deep Learning Networks For Accurate Prediction Of Relative Positions Of Remote Vehicles

4.4.1 LSTM Network Models for Prediction of Relative Position of a remote vehicle

We applied the same VMP-L system on V2V dataset for remote vehicle position prediction as shown in Fig. 24.

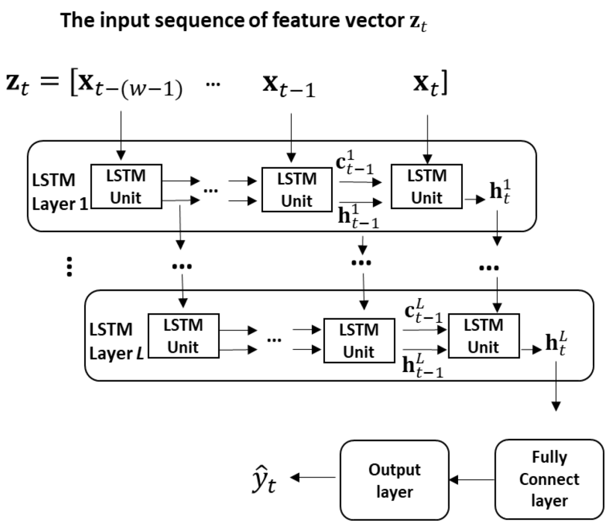


Figure 25. Architecture of a multilayered LSTM network with input features extracted from V2V communication data and output the position of a remote vehicle.

4.4.2 Attention focused transformer models for prediction of relative position of a remote vehicle

We also evaluate VMP-T system on V2V dataset for remote vehicle position prediction.

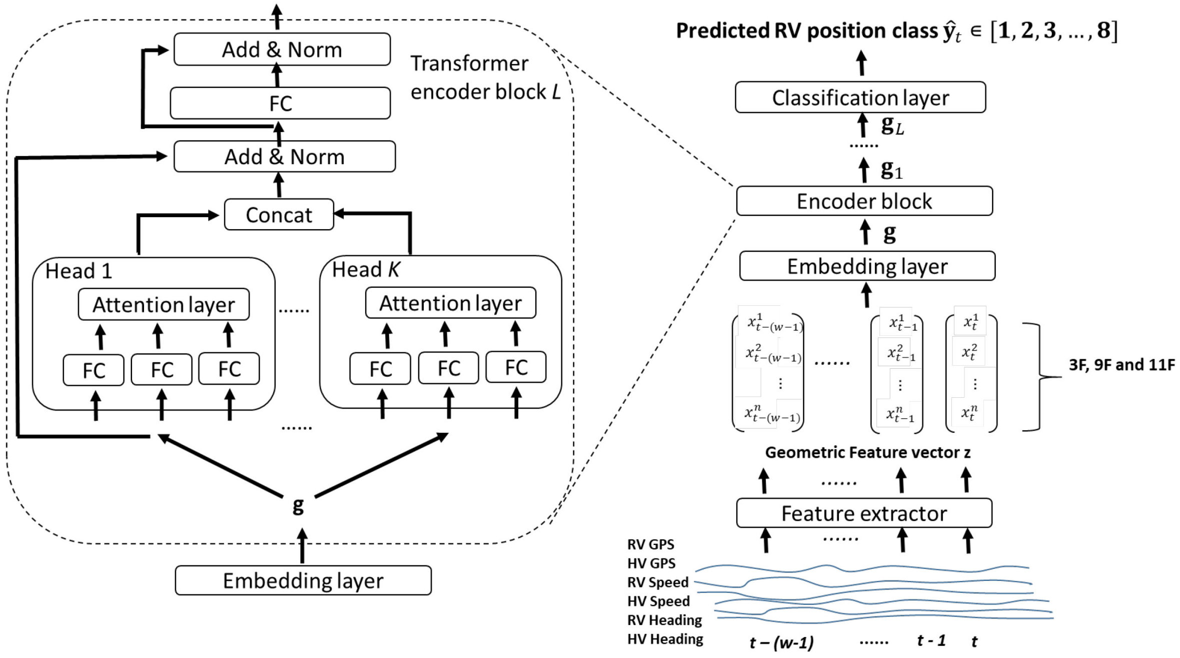


Figure 26. VMP-T: an attention focused transformer network for remote vehicle position prediction.

4.5 Experiments Setup

In this section, we evaluate the proposed geometric features extracted from V2V signals and deep learning models for detecting and predicting nearby remote vehicle positions using V2V data collected from naturalistic driving trips.

4.5.1 Hyper-parameter settings

In the experiments that follow, we used the following hyper-parameter settings:

- Geometric features: 3F, 9F, and 11F
- Data sampling rate: 10 Hz
- Window size: $w = 5$ for LSTM / VMP-T, $w = 10$ for MLP
- Number of output classes: $K = 8$
- Max number of training epochs: $epochs = 100$
- Number of LSTM layers: $L = 1,2,3$
- Number of hidden units in each LSTM layer: k_1 (first layer) varies, $k_2 = 0.5k_1$, $k_3 = 0.5k_2$.
- The dimension of VMP-T embedding layer: 126
- Look ahead prediction time: $\Delta t = 0.1, \dots, 1.0, 2.0, 3.0$ seconds.

4.5.2 V2V data used in experiments

We used the V2V data recorded by in real world driving. The dataset was collected by the University of Michigan Transportation Research Institute (UMTRI) under the program: Safety Pilot Model Deployment (SPMD), sponsored by the Department of Transportation (DOT), USA [77]. The SPMD program involved installing dedicated short-range communication (DSRC) devices operating at 5.9 GHz on over 1800 vehicles in and around Ann Arbor, Michigan, USA, over 23 months, beginning August, 2012. All DSRC devices sent the basic safety message (BSM) at 10 HZ on all vehicles, which included information about the vehicle that sent it such as its GPS position, speed, and heading [78].

The BSMs collected for each vehicle trip contained the following information:

1. Trip Id
2. Vehicle Id.
3. Time stamp
4. Vehicle speed (m/s)
5. GPS Latitude (5 decimal digits)
6. GPS Longitude

Our research focuses on the trips that host vehicle and remote vehicle are within 10 m of each other. We extracted 69 trip segments that met this requirement in total, amounting to 50% more trips than what was used in [72]. Trip segments includes various driving scenarios like city and highway driving on straight and curved roads.

Fig.26 shows a portion of one such trip. The trajectory of the host vehicle is shown in red circles and the trajectory of the remote vehicle in blue circles. The two vehicles are driving in opposite directions, as indicated by the arrows. We down sampled the original data to have a frequency of 1 HZ. The positional changes of the remote vehicle relative to the host vehicle are: class 1 → 4 → 6.

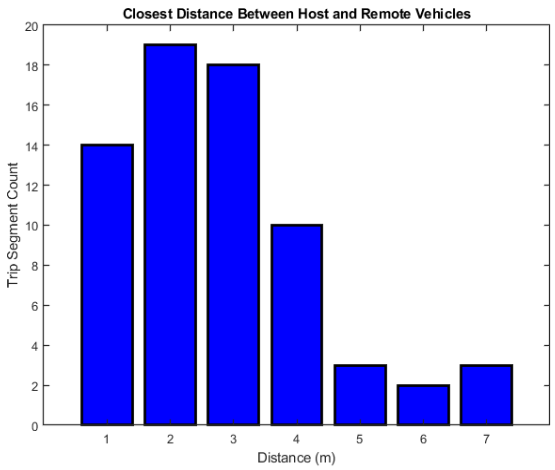


Figure 27. A histogram of the closest distances between the pairs of a host and a remote vehicle among experimental data.

Figures 27 – 29 show statistics of the extracted trip segments. Figure 27 summarizes the closest distances between the host and the remote vehicles in the dataset used in our experiments. In most trips, the vehicles come within less than 4m of each other.

The histogram in Figure 10 shows how long the host and remote vehicles are close to each other. The cases with very short proximity times usually happen when the host and remote vehicles are driving in opposite directions. About 20% of the 69 trip segments, or 14 cases, have opposite driving. the host and remote vehicles are driving in opposite directions. Note that opposite driving

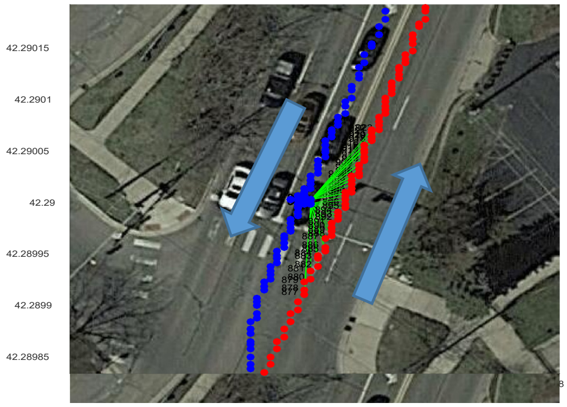


Figure 28. An example shows the relative positions between a host and a remote vehicle in one trip segment. The positions of the host vehicle are in red, and those of the remote vehicle are in blue. Two vehicles traveled in the directions indicated by the blue arrows.

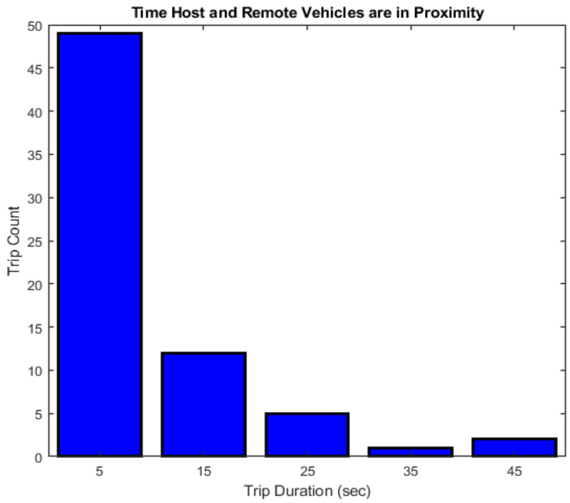


Figure 29. Statistics of duration of proximity between the host vehicle and the remote vehicle.

Figure 29 is a bar graph illustrating the numbers of data samples available in each of the 8 vehicle position classes. It is evident that this data set is not balanced, with class 8 containing the highest number of samples and classes 2 and 4 having comparatively fewer. The detailed description of data preparation and obtaining the ground truth of the vehicle position classes in the data can be found in [72].

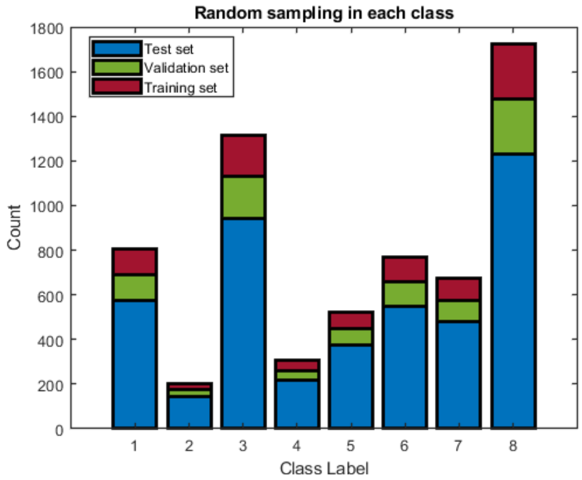


Figure 30. Number of data samples in each class. The blue segment represents the fraction of data designated for training during random sampling. The green and red segments depict the proportions assigned to the validation and test sets respectively.

The workflow of remote vehicle position prediction using V2V data is shown in Fig. 30.

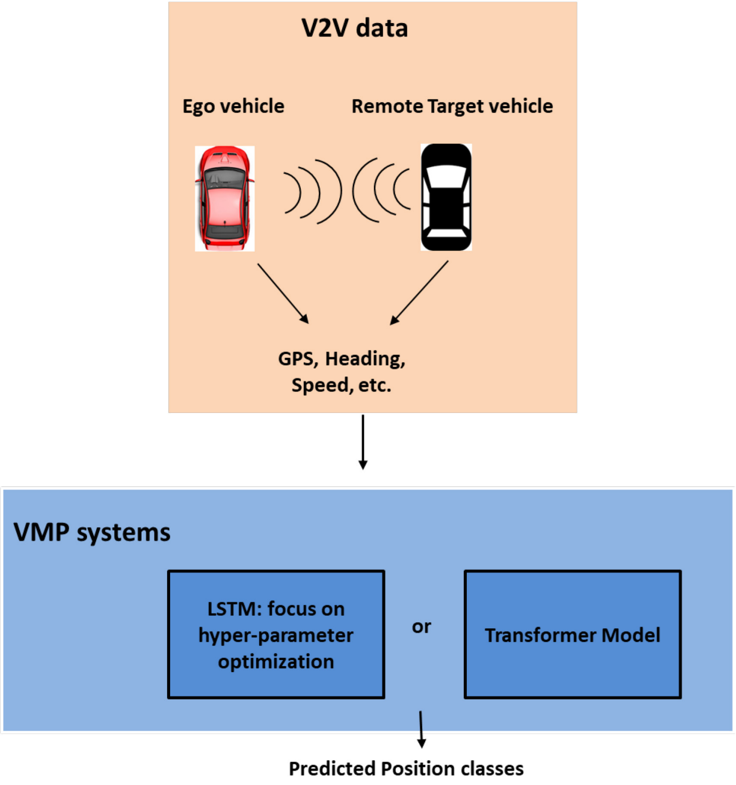


Figure 31. Remote vehicle position prediction using V2V data.

4.5.3 Three Evaluation Methods

In this study, we used the following three methods to evaluate the proposed machine learning models.

- Method 1: The data is partitioned into 70% for the training set, 15% for the validation and 15% for the test set using the stratified sampling method. Systems were trained using training and validation data, and performances are evaluated on the test data.
- Method 2: Leave-one-out (complete trip) training and test method.
- Method 3. Select four representative trips as test data for system performance evaluation.

The last two methods use trip-based performance analysis. Since we have a limited number of trip segment data, (69 in total), method 2 uses a leave-one-out strategy to evaluate the proposed systems. In this case, 69 rounds of tests were conducted. At every round, one complete trip segment was used as a test set, and the rest 68 trip segments were used for training. No training and test data in two rounds are identical. The system performance is measured by averaging performance over all 69 test sets. It should be noted that the leave-one-out approach is a specific instance of K-fold testing, in which $K = N_T - 1$ and N_T represents the total number of test trips.

For the third method, four representative trip segments were selected from the 69 trip segments as the test set, and the rest 65 trip segments were used as a training set. The four test trips were chosen such that, two trips in which the HV and RV were traveling in the same direction, and in the other two trip segments HV and RV were traveling in the opposite direction, and each had a good representation of the output classes. Table 4.1 shows the statistics of 4 test trip segments. In total, there are 92 data samples extracted from the four test trips.

The RV position prediction performances of the proposed systems are measured by accuracy on test data defined by:

$$Accuracy = \frac{\#\, Samples\ Correctly\ Classified}{Total\ \#\, Samples\ Tested}$$

Table 4.1 Statistics of the test set containing 4 complete trips

Test Trip	**HV Speed (m/sec)**	**RV Speed (m/sec)**	**Closest Distance (m)**	**Output Classes**	**Traveling Direction**
1	8.92	10.67	3.45	1, 4, 6	Opposite
2	4.33	8.31	4.23	1, 4, 6	Opposite
3	5.06	14.00	4.83	3, 5, 8	Same
4	14.28	18.33	3.54	1, 4, 6	Same

In this section we present the performance analyses on the systems developed for remote vehicle relative position prediction. The system performances are measured in accuracy. It should be noted that the vehicle relative position prediction is made at time *t* for vehicle relative position at time $t + \Delta t$. As the prediction horizon increases, the data samples used in prediction require longer historical time window, therefore, short trips cannot be used. Table 4.2 summaries the number of available trips using prediction horizon Δt in the range: 0.1 ~ 3 seconds, and the number of valid trips used in the respective experiments.

Table 4.2 Numbers of trips for different predict time

Predict Time (seconds)	0.1	0.2	0.3	0.4	0.5	1	2	3
#Trips	67	67	65	65	63	57	47	43

4.6 Experiment results

The prediction results generated by MLP, VMP-L and VMP-T systems are presented in Tables 4.3 ~ 4.6. These tables show the accuracies of relative position predictions by VMP-T, LSTM and MLP systems using feature sets: 3F, 9F, and 11F. The predict-ahead time is in the range of 0.1s to 0.5s by an increment of 0.1s, as well as the longer prediction time horizons of 1, 2, and 3 seconds. In order to provide succinct reports in these tables, we presented average prediction results between 0.1 ~ 0.4s and 0.5 ~ 1s, and at 2 and 3s.

As expected, the prediction accuracies, in general, decrease as the prediction horizon increases. The 3F VMP-L system gave the highest accuracies in comparison to the MLP and VMP-T; in particular, when the prediction time horizons are small. When feature sets 9F and 11F were used, both the VMP-T and LSTM systems gave better prediction results than the systems using 3F. The proposed transformer-based system, VMP-T and the LSTM-based VMP-L systems outperform the MLP systems in all experiments by large margins. The transformer systems using 11F achieved the highest prediction accuracies among all other systems at every predict-ahead time instance. On average, 11F VMP-T outperformed the 11F MLP system by 18%.

Table 4.3 Overall performances of systems using 3F by averaging three methods

3 Features (3F)		Remote vehicle prediction accuracy (%)		
		MLP	VMP-L	VMP-T
Δt(s)	0.1-0.4	85.0	89.4	84.0
	0.5-1	78.5	86.0	80.7
	2	62.4	75.4	75.8
	3	54.3	71.3	71.6
Average		**70.1**	**80.5**	**78.0**
Std.		12.3	7.4	4.7

Table 4.4 Overall performances of systems using 9F by averaging three methods

9 Features (9F)		Remote vehicle prediction accuracy (%)		
		MLP	VMP-L	VMP-T
Δt(s)	0.1-0.4	83.9	88.8	90.1
	0.5-1	77.9	85.0	87.8
	2	63.6	76.4	86.6
	3	55	72.5	83.6
Average		**70.1**	**80.7**	**87.0**
Std.		11.4	6.5	2.3

Table 4.5 Overall performances of systems using 11F by averaging three methods

11 Features (11F)		Remote vehicle prediction accuracy (%)		
		MLP	VMP-L	VMP-T
Δt(s)	0.1-0.4	81.1	88.3	91.3
	0.5-1	78.0	85.4	89.6
	2	65.1	76.3	87.0
	3	55.5	72.9	83.8
Average		**69.9**	**80.7**	**87.9**
Std.		10.3	6.3	2.8

Table 4.6 Comparison of effective features used in RV position prediction at various prediction horizons

		Average performances using different features		
		3F	9F	11F
Δt(s)	0.1-0.4	86.0	87.5	86.8
	0.5-1	81.6	83.9	84.3
	2	71.6	76.2	76.8
	3	67.0	71.2	71.6
AVERAGE		**76.6**	**79.7**	**79.9**
STD.		7.6	6.4	6.0

4.7 Conclusion

We proposed two DLNN models: LSTM and VMP-T and three sets of effective geometric features for accurate prediction of eight classes of remote vehicle positions based on V2V communication signals. We evaluated these deep learning models and conducted comparative studies with MLP-based systems through extensive experiments using a set of V2V communication data collected from 69 naturalistic driving trips. System performance was evaluated using a combination of three methods: random sampling, leave-one-out, and a test set of four representative trips. Our findings are as follows.

- For the remote vehicle position predictions, the proposed transformer-based system, 11F VMP-T gave the state-of-art performance: prediction accuracy reached 91.3% when prediction time is in the range 0.1s ~ 0.4s, 89.6 when the prediction time is 0.5s and 1s, 87% and 83.8% when the prediction time is 2s and 3s, respectively.
- The average prediction accuracy of the 11F VMP-T system over all prediction horizons improved 17.8% over the best MLP system, and 7.2% over the best VMP-L system.
- We explored the effectiveness of three different feature vectors, 3F, 9F and 11F, for remote vehicle position prediction. 3F is a compact feature vector, containing three primary geometric

features, 9F contains 9 features obtained by augmenting the primary geometric features with spatial position information of the host and remote vehicles, and 11F contains 11 features obtained by combining the features in 9F with the speed of the host and remote vehicles at t. When MLP is used, 3F provides competitive performances over 9F and 11F in terms of prediction accuracy and computational efficiency. Similarly 3F VMP-L performed competitively with 9F and 11F VMP-L in prediction. However, the systems used 11F as input features and VMP-T as prediction models gave the best prediction accuracy over all other systems.

Both the proposed transformer based system, VMP-T, and LSTM based system, VMP-L outperformed MLP systems by large margins in remote vehicle position the prediction. The VMP-T systems with input feature vector of either 9F or 11F outperformed all LSTM systems.

Chapter 5: Conclusions and Future Directions

In this dissertation, we have presented a comprehensive study of vehicle maneuver prediction for both ego and remote target vehicle using deep learning models. We began by reviewing the literature on vehicle maneuver prediction, highlighting the limitations of traditional machine learning models and the potential advantages of deep learning models. We then presented our proposed approaches for vehicle maneuver prediction using a combination of CNNs, LSTMs, and attention mechanisms, and demonstrated its effectiveness on different kinds of real-world driving data. We conducted ablation experiments to study the how temporal information and visual information improve the performances of VMP systems.

In this research, we focus on studying two different aspects of vehicle maneuver prediction: ego vehicle maneuver prediction based on in-vehicle sensor data, and remote target vehicle maneuver prediction using V2V and V2I data.

For the first aspect, we propose a deep learning-based approach to predict the future movements of the ego vehicle based on information provided by in-vehicle sensors' data, such as videos, GPS, speed and headings. Our approach utilizes a combination of CNN, attention networks to extract features from sensor data and model the temporal dependencies in the data. We evaluate our proposed approach on real-world driving data and demonstrate its effectiveness in accurately predicting the future movements of the ego vehicle.

For the second aspect, we propose two different approaches for remote target vehicle maneuver prediction. The first approach utilizes V2V communication to exchange information between neighboring vehicles, and the second approach utilizes V2I communication to receive information from traffic signals and other infrastructure. Both approaches use a combination of deep learning models and sensor data to predict the future maneuvers or positions of remote target vehicles. We evaluate both approaches on real-world driving data and demonstrate their effectiveness in accurately predicting the future movements of remote target vehicles.

While our proposed approaches for vehicle maneuver prediction have shown promising results, there is still much room for improvement and further research in this area. In particular, future work may focus on the following areas:

1. Incorporating additional types of input data: While our proposed approaches incorporate visual data, sensor data, and contextual information, there may be other types of input data that could improve the performance of the models, such as Lidar and radar data.

   Lidar and radar data can bring several important benefits to vehicle maneuver prediction systems. Firstly, Lidar and radar data can provide accurate measurements of the distance between the ego vehicle and other vehicles or objects in the environment. This information can be used to detect potential collision risks and predict the future movements of other vehicles. Lidar and radar data can also provide information on the velocity and direction of other vehicles, which is essential for predicting their future movements. Secondly, Lidar and radar data can provide a more comprehensive view of the environment, especially in challenging weather conditions where visual data may be limited. This allows for more accurate and reliable vehicle maneuver prediction, even in adverse driving conditions. Thirdly, Lidar and radar data can be used in conjunction with other types of input data, such as visual data and GPS data, to provide a more accurate and comprehensive view of the environment. By combining data from multiple sources, vehicle maneuver prediction systems can improve their accuracy and reliability, making them more effective in real-world driving scenarios.

Overall, the use of Lidar and radar data in vehicle maneuver prediction systems can improve their performance and reliability, enabling the development of more advanced autonomous driving systems that can operate in a wider range of driving scenarios. Further research in this area may focus on developing more advanced sensor fusion techniques to combine data from multiple sources and improve the accuracy and reliability of vehicle maneuver prediction systems.

2. Evaluating the performance in diverse driving scenarios: Our proposed approaches have been evaluated on real-world driving data, but there may be other driving scenarios, such as broader rural or urban environments, adverse weather conditions, that could pose unique challenges for vehicle maneuver prediction. Future work may focus on evaluating the performance of the models in a wider range of driving scenarios.
3. Evaluating the performance in real-time systems: While our proposed approaches have been demonstrated offline on pre-recorded data, there may be challenges associated with implementing the models in real-time on a moving vehicle. Future work may focus on developing real-time implementations of the models that can be used in autonomous driving systems.

Overall, further research in this area has the potential to significantly improve the safety and reliability of autonomous driving systems, enabling them to operate more effectively and accurately in a wider range of driving scenarios.

References

[1] T. Stewart, "Overview of motor vehicle crashes in 2020," Report No. DOT HS 813266, National Highway Traffic Safety Administration, March 2022.

[2] Y. Zheng and J. H. L. Hansen, "Lane-change detection from steering signal using spectral segmentation and learning-based classification," IEEE Transactions on Intelligent Vehicles, vol. 2, pp. 14–24, 2017.

[3] A. Zyner, S. Worrall, J. Ward, and E. Nebot, "Long short term memory for driver intent prediction," in 2017 IEEE Intelligent Vehicles Symposium (IV), June 2017, pp. 1484–1489.

[4] Insurance Institute for Highway Safety, https://www.iihs.org/news/detail/front-crash-prevention-slashes-police-reported-rear-end-crashes#:~:text=Forward%20collision%20warning%20alone%20reduced,U.S.%20police%2Dreported%20crash%20data, last visited on April 20th, 2022.

[5] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," in IEEE Inter. Conf. on Intel. Computer Commun. and Proc. (ICCP), 2009, pp. 417–422.

[6] G.S. Aoude, B.D. Luders, K.K. Lee, D.S. Levine and J.P. How, "September. Threat assessment design for driver assistance system at intersections", in 13th international IEEE conference on intelligent transportation systems (pp. 1855-1862), 2010.

[7] G.S. Aoude, V.R. Desaraju, L.H. Stephens and J.P. How, “Driver behavior classification at intersections and validation on large naturalistic data set”, in IEEE Transactions on Intelligent Transportation Systems, 13(2), pp.724-736, 2012.

[8] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, “Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning,” in Conference on Computer Vision and Pattern Recognition, 2018.

[9] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-V3, inception-resnet and the impact of residual connections on learning.” In AAAI, vol. 4, 2017, p. 12.

[10] H. Xu, Y. Gao, F. Yu, T. Darrell, End-to-end learning of driving models from large-scale video datasets, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2174-2182.

[11] D. Lenz, F. Diehl, M. T. Le, and A. Knoll, “Deep neural networks for markovian interactive scene prediction in highway scenarios,” in 2017 IEEE Intelligent Vehicles Symposium (IV), Jun. 2017, pp. 685–692.

[12] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, “Imitating driver behavior with generative adversarial networks,” in 2017 IEEE Intelligent Vehicles Symposium (IV), Jun. 2017, pp. 204–211.

[13] S. Dai, L. Li and Z. Li. “Modeling vehicle interactions via modified LSTM models for trajectory prediction.” In IEEE Access, 7, pp.38287-38296, 2019.

[14] M. Gori, G. Monfardini, and F. Scarselli, “A new model for learning in graph domains,” in Proc. IEEE Int. Joint Conf. Neural Netw., Jul. 2005, pp. 729–734.

[15] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” IEEE Trans. Neural Netw., vol. 20, no. 1, pp. 61–80, Jan. 2009.

[16] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” arXiv preprint arXiv: 1609.02907, 2016.

[17] F. Diehl, T. Brunner, M. T. Le, and A. Knoll, “Graph neural networks for modelling traffic participant interaction,” in Proc. IEEE Intell. Vehicles Symp., Jun. 2019, pp. 695–701.

[18] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in International Conference on Machine Learning, 2013, pp. 1310–1318.

[19] J. Kim and J. F. Canny, “Interpretable learning for self-driving cars by visualizing causal attention.” in ICCV, 2017, pp. 2961–2969.

[20] O. Olabiyi, E. Martinson, V. Chintalapudi, and R. Guo, “Driver action prediction using deep (bidirectional) recurrent neural network,” arXiv preprint arXiv: 1706.02257, 2017.

[24] S. Wang and Y. L. Murphey, "Driving Maneuver Detection using Features of Driver's attention and Face Shift through Deeping Learning," 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9534110.

[25] A. Palazzi, D. Abati, S. Calderara, F. Solera, and R. Cucchiara, “Predicting the driver’s focus of attention: the dr(eye)ve project,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.

[26] Y. Xia, D. Zhang, A. Pozdnoukhov, K. Nakayama, K. Zipser, and D. Whitney, “Training a network to attend like human drivers saves it from common but misleading loss functions,” CoRR, vol. abs/1711.06406,2017.

[27] X. Peng, A. Zhao, S. Wang, Y. Murphey and Y. Li. “Attention-Driven Driving Maneuver Detection System.” In International Joint Conference on Neural Networks (IJCNN) 2019.

[28] S. Sharma, R. Kiros, and R. Salakhutdinov. “Action recognition using visual attention”. ICLR, 2016.

[29] S. Alletto, A. Palazzi, F. Solera, S. Calderara and R. Cucchiara, "DR(eye)VE: A Dataset for Attention-Based Tasks with Applications to Autonomous and Assisted Driving," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, 2016, pp. 54-60, doi: 10.1109/CVPRW.2016.14.

[30] J. Colyar and J. Halkias, “Us highway 80 dataset,” Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-07-030, 2007.

[31] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition (ICPR), pp. 770-778, in 2016.

[32] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv: 1409.1556, in 2014.

[33] D. Lee, Y. P. Kwon, S. Mcmains, and J. K. Hedrick, “Convolution neural network-based lane change intention prediction of surrounding vehicles for ACC,” in Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC), Oct. 2017, pp. 1–6.

[34] H. Cui et al., “Multimodal trajectory predictions for autonomous driving using deep convolutional networks,” in Proc. Int. Conf. Robot. Automat. (ICRA), 2019, pp. 2090–2096.

[35] A. Zyner, S. Worrall, and E. Nebot, “A recurrent neural network solution for predicting driver intention at unsignalized intersections,” IEEE Robot. Autom. Lett., vol. 3, no. 3, pp. 1759–1764, Jul. 2018.

[36] D. J. Phillips, T. A. Wheeler, and M. J. Kochenderfer, “Generalizable intention prediction of human drivers at intersections,” in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2017, pp. 1665–1670.

[37] T. Baltrusaitis, P. Robinson, and L-P. Morency. “Constrained local neural fields for robust facial landmark detection in the wild”. In ICCV Workshop, 2013.

[38] A. Zadeh, T. Baltrušaitis and L. Morency, "Convolutional Experts Constrained Local Model for Facial Landmark Detection," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017.

[39] X. Li, X. Ying, and M. C. Chuah, “Grip: Graph-based interaction-aware trajectory prediction,” in 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 3960–3966.

[40] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, “Recurrent neural networks for driver activity anticipation via sensory-fusion architecture,” in 2016 IEEE International Conference on Robotics and Automation (ICRA), May 2016, pp. 3118–3125.

[41] K. Cho et al., “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1724–1734.

[42] W. Ding, J. Chen, and S. Shen, “Predicting vehicle behaviors over an extended horizon using behavior interaction network,” in International Conference on Robotics and Automation (ICRA), 2019, pp. 8634–8640.

[43] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, “Trafficpredict: Trajectory prediction for heterogeneous traffic-agents,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2pp. 6120–6127, 2019.

[44] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, “Multi-agent tensor fusion for contextual trajectory prediction,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[45] A. Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, “Attention Is All You Need,” 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

[46] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly and J. Uszkoreit,. “An image is worth 16x16 words: Transformers for image recognition at scale”. In International Conference on Learning Representations (ICLR), 2021.

[47] Quintanar, A., David Fernández Llorca, Ignacio Parra, Rubén Izquierdo and Miguel Ángel Sotelo. “Predicting Vehicles Trajectories in Urban Scenarios with Transformer Networks and Augmented Information.” 2021 IEEE Intelligent Vehicles Symposium (IV) (2021): 1051-1056.

[48] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, “Transformer Networks for Trajectory Forecasting.” 25th International Conference on Pattern Recognition (ICPR), 2020.

[49] N. Djuric et al., “Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2020.

[50] N. Deo and M. M. Trivedi, “Convolutional social pooling for vehicle trajectory prediction,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2018, pp. 1468–1476.

[51] Alzubaidi, L., Zhang, J., Humaidi, A.J. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8.

[52] W. Luo, B. Yang, and R. Urtasun, “Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3569–3577.

[53] X. Peng, X., Murphey, Y.L., Liu, R. and Li, Y., 2020. Driving maneuver early detection via sequence learning from vehicle signals and video images. Pattern Recognition, 103, p.107276.

[54] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015).

[55] Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. "Show, attend and tell: Neural image caption generation with visual attention." In International conference on machine learning, pp. 2048-2057. PMLR, 2015.

[56] L. Tijerina, W. Garrott, D. Stoltzfus, and E. Parmer, “Eye glance behavior of van and passenger car drivers during lane change decision phase,” Transportation Research Record: Journal of the Transportation Research Board, pp. 37–43, 2005.

[57] Fernando, Tharindu, Simon Denman, Sridha Sridharan, and Clinton Fookes. "Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection." Neural networks 108 (2018): 466-478.

[58] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet and F. Nashashibi, "Attention Based Vehicle Trajectory Prediction," in IEEE Transactions on Intelligent Vehicles, vol. 6, no. 1, pp. 175-185, March 2021, doi: 10.1109/TIV.2020.2991952.

[59] H. Kim, D. Kim, G. Kim, J. Cho and K. Huh, "Multi-Head Attention based Probabilistic Vehicle Trajectory Prediction," 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 2020, pp. 1720-1725, doi: 10.1109/IV47402.2020.9304741.

[60] K. Messaoud, N. Deo, M. M. Trivedi and F. Nashashibi, "Trajectory Prediction for Autonomous Driving based on Multi-Head Attention with Joint Agent-Map Representation," 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 2021, pp. 165-170, doi: 10.1109/IV48863.2021.9576054.

[61] B. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei. “Imagenet: A large-scale hierarchical image database,” In Computer Vision and PatternRecognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 248-255.

[62] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. IVC, 28(5):807 – 813, 20.

[63] Yi Lu Murphey, Ilya Kolmanovsky, and Paul Watta, Editors, AI-enabled Technologies for Autonomous and Connected Vehicles, Springer, July, 2022.

[64] Y. Shen, H. Dong, L. Jia, Y. Qin, F. Su, M. Wu, K. Liu, P. Li, Z. Tian, “A Method of Traffic Travel Status Segmentation Based on Position Trajectories.” Proceedings of the IEEE 18th International Conference on Intelligent Transportation Systems (ITSC), 2015.

[65] R. Zhang, L. Cao, S. Bao, and J. Tan, “A method for connected vehicle trajectory prediction and collision warning algorithm based on V2V communication,” International Journal of Crashworthiness, 22(1), 15-25, 2017.

[66] S. Patra, C. Calafate, J. Cano, C. de Vera, P. Veeleart, and W. Philips, “Determining the relative position of vehicles considering bidirectional traffic scenarios in VANETS,” SmartObjects '16 Proceedings of the 2nd Workshop on Experiences in the Design and Implementation of Smart Objects, Pages 6-10, New York City, New York — October 03 - 07, 2016.

[67] C. Barrios and Y. Motai, “Improving Estimation of Vehicle's Trajectory Using the Latest Global Positioning System With Kalman Filtering.” IEEE Trans. Instrumentation and Measurement. 60, 3747-3755, 2011.

[68] A. Bhawiyuga, H. Nguyen, H. Jeong, “An Accurate Vehicle Positioning Algorithm based on Vehicle-to-Vehicle (V2V) Communications,” Proceedings of the IEEE International Conference on Vehicular Electronics and Safety, Istanbul, Turkey, July 24-27, 2012.

[69] H. Cho and B. Kim, “Study on Cooperative Intersection Collision Detection System Based on Vehicle-to-Vehicle Communication,” Advanced Science and Technology Letters, Vol. 58, pp.121-124, 2014.

[70] J. Harding, G. R. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons, and J. Wang, “Vehicle-to-vehicle communications: Readiness of V2V technology for application.”

(Report No. DOT HS 812 014). Washington, DC: National Highway Traffic Safety Administration, 2014.

[71] M. El-Said, S. Mansour, V. Bhuse, “DSRC Based Sensor-Pooling Protocol for Connected Vehicles in Future Smart Cities,” Procedia Computer Science, Volume 140, pp. 70-78, 2018.

[72] P. Watta, X. Zhang, and Y. L. Murphey, “Vehicle Position and Context Detection using V2V Communication.” IEEE Transactions on Intelligent Vehicles, Volume 6, Issue: 4, Dec. 2021.

[73] Koon, John. “Will Vehicle-to-Vehicle Communication Ever Take Off?” Engineering, 20 February 2019. https://www.engineering.com/story/will-vehicle-to-vehicle-communication-ever-take-off, last visited 1/7/2023.

[74] J. Gao, Jiangang Yi and Yi Lu Murphey, “Attention-based global context network for driving maneuvers prediction,” Machine Vision and Applications (2022) 33:53,

[75] A. Zyner, S. Worrall, and E. M. Nebot, “Naturalistic driver intention and path prediction using recurrent neural networks,” in IEEE transactions on intelligent transportation systems 21, no. 4 (2019): 1584-1594..

[76] L. Xin, P. Wang, C. Chan, J. Chen, S. E. Li, and B. Cheng, “Intention aware long horizon trajectory prediction of surrounding vehicles using dual LSTM networks,” Proceedings of the International Conference on Intelligent Transportation Systems (ITSC), pp. 1441–1469, November 2018.

[77] “Questions and Answers about DOT’s Safety Pilot: Model Deployment.” [Online]. Available:https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/technical_fact_sheet-model_deployment.pdf. Last visited January 14th, 2023.

[78] Y. Liu, P. Watta, J. Bochen, and Y. L. Murphey, Y. Vehicle position and context detection using V2V communication with application to pre-crash detection and warning," Proceedings of the 2016 IEEE Symposium on Computational Intelligence, Athens, Greece, December 6 – 9, 2016.