ON THE PERFORMANCE OF DEEP LEARNING MODELS IN ENHANCED C-V2X COMMUNICATION

by

KHALED KORD

A thesis submitted to the

School of Computing

in conformity with the requirements for

the degree of Master of Science

Queen's University

Kingston, Ontario, Canada

December 2021

Abstract

Autonomous vehicles (AVs) are an essential component of future intelligent transportation systems (ITS). AVs promise an increase in driving comfort, greater safety, and improved fuel consumption. These benefits are attributed to AVs ability to leverage vehicle-to-everything (V2X) communication to coordinate driving, e.g., sharing maneuver intention and acquiring updated road maps. Cellular based V2X (C-V2X) had been popularized as one of the leading technologies at the forefront of ITS. C-V2X technology is already capable of supporting a basic set of AVs' uses cases, e.g. warning messages for collision avoidance. In contrast, support for advanced use cases, e.g. cooperative lane change, is still an obstacle. Advanced AV use cases require a network able to support up to 10ms end-to-end (E2E) latency and 99.99% packet delivery rate. These stringent service requirements pose a challenge for 4G LTE and 5G New Radio (NR). In this work, we build a simulation environment to study the feasibility of 5G NR in supporting advanced use cases in AVs and propose deep learning-based solutions to enhance its performance further. First, we propose deep leaning models to perform uplink channel state information (CSI) prediction in dynamic vehicular environments. The models are based on a combination of convolutional and recurrent neural networks (CNN-RNN), so as to suit the mobility factor in vehicular environments. Next, we propose a novel scheme based on deep learning prediction to enhance

the uplink resource allocation process in 5G C-V2X. The proposed scheme enables the base station to predict vehicle maneuvers, subsequently, assign it the required resource in advance without the need for scheduling request and resource granting process. We show that this scheme improves the ability of 5G NR to support cooperative driving requirements. Moreover, we compare to traditional schemes discussing issues that arise from the introduction of prediction models and possible approaches for further enhancements in the future.

Acknowledgments

First of all, I would like to express my gratitude to Prof. Sameh Sorour, my supervisor who sadly passed away before the final submission of this work. Prof. Sorour was a true embodiment of hard work, and I will always be thankful for the unrelenting support he gave me. I would also like to thank Prof. Hossam Hassanein who assumed the role of acting supervisor in a difficult time. Special thanks and gratitude should also be given to Dr. Ahmed Elbery for the great intuitions, ideas, and for the many hours he spent caring for every detail. I thank Prof. Hazem Abbas for his belief in me. I also thank Ziyad Hatem and Mohamed Mossad for encouraging and supporting me when I needed it the most. I also thank my family as without them I wouldn't be half the man I'm today. I gained many friends throughout the journey that is postgraduate studies in Queen's. I would like to thank Bassel Chawky, my fellow student at Queen's Castle lab for hours of banter, games of chess. I would also like to express gratitude for Shaza Kaoud, Ahmed Nagib and Rawan El-Khatib for their help with non-academic matters when I first arrived at Canada as a foreigner. I also thank my flatmate, Lee Zhang for the great food, cheerful spirit and for making Kingston feel like home. Finally, I thank Amr Abu-talib and Sherif Azmy for their valuable advice regarding the writing of my thesis.

List of Publications

1. K. Kord, A. Elbery, S. Sorour and H. S. Hassanein, "On the Performance of Deep Learning Models for Uplink CSI Prediction in Vehicular Environments," ICC 2021 - IEEE International Conference on Communications, 2021, pp. 1-6, doi: 10.1109/ICC42927.2021.9500269.

2. K. Kord, A. Elbery, S. Sorour and H. S. Hassanein, Akram Bin Sediq, Ali Afana, Hatem Abou-zeid "Enhanced C-V2X Uplink Resource Allocation using Vehicle Maneuver Prediction," Submitted to IEEE International Conference on Communications - ICC 2022.

Abbreviations

3GPP The 3rd Generation Partnership Project

ADR Average Difference Ratio

AV Autonomous Vehicle

BS Base Station

BSR Buffer Status Report

C-V2X Cellular Vehicle-to-Everything

CNN Convolutional Neural Network

CSI Channel State Information

CSI-RS Channel State Information Reference Signal

D2D Device-to-Device Communication

DCI Download Control Information

DL Downlink

DMRS Demodulation Reference Signal

DRSC Dedicated Short-range Communication

ITS Intelligent Transportation Systems

LOS Line of Sight

MIMO Multiple-Input Multiple-Output

ML Machine Learning

MLP Multi Layer Perceptron

MSE Mean Square Error

NR 5G New Radio

OFDM Orthogonal Frequency-division Multiplexing

OSM OpenStreetMap

PDCCH Physical Downlink Control Channel

PDSCH Physical Downlink Shared Channel

PSFCH Physical Sidelink Feedback Channel

PTRS Phase-tracking Reference Signal

PUCCH Physical Uplink Control Channel

PUSCH Physical Uplink Shared Channel

QoS Quality of Service

RB Resource Block

ReLU Rectified Linear Unit

RMa Rural Macrocell

RNN Recurrent Neural Network

SR Scheduling Request

SRS Sounding Reference Signal

SUMO Simulation of Urban Mobility Tool

UCI Uplink Control Information

UE User Equipment

UL Uplink

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

V2X Vehicle-to-Everything

Contents

Abstract i

Acknowledgments iii

List of Publications iv

Contents viii

List of Tables x

List of Figures xi

Chapter 1: Introduction 1
- 1.1 Overview and Motivation . . . 1
- 1.2 C-V2X Challenges . . . 2
  - 1.2.1 Security . . . 2
  - 1.2.2 Infrastructural and Economic Considerations . . . 3
  - 1.2.3 Performance . . . 4
- 1.3 Objective and Contributions . . . 5
- 1.4 Organization of Thesis . . . 7

Chapter 2: C-V2X Overview 8
- 2.1 Physical Layer . . . 8
  - 2.1.1 Frame structure . . . 8
  - 2.1.2 Modulation and Control Channels . . . 9
  - 2.1.3 Channel Estimation and Reference Signals . . . 10
- 2.2 C-V2X Scheduling . . . 11
- 2.3 V2X Interfaces . . . 13
  - 2.3.1 Uu Interface . . . 13
  - 2.3.2 PC5 Interface . . . 14
- 2.4 C-V2X Use Cases . . . 15
- 2.5 Literature Review . . . 16

Chapter 3: CSI Prediction **19**
3.1 Problem Formulation . . . 19
3.2 Data Generation . . . 20
3.2.1 Data Generation Setup . . . 21
3.2.2 The Dataset . . . 22
3.3 Model Architectures . . . 23
3.3.1 MLP Model . . . 24
3.3.2 CNN-RNN Model . . . 26
3.4 Sensitivity Analysis . . . 29
3.4.1 Sensitivity to Speed . . . 29
3.4.2 Sensitivity to History and Prediction Horizon . . . 31
3.5 Results Discussion . . . 32
3.5.1 Comparison of the Two Models . . . 32
3.5.2 Comparison with State of the Art . . . 33

Chapter 4: Maneuver Prediction **34**
4.1 Problem Formulation . . . 34
4.2 Dataset . . . 37
4.2.1 Mobility Dataset Generation Method . . . 37
4.2.2 NGSIM . . . 39
4.3 Prediction Model . . . 41
4.3.1 Performance of Different Classifiers . . . 41
4.3.2 LSTM Model . . . 42
4.3.3 Implementation . . . 42
4.3.4 Feature Importance . . . 44
4.3.5 Effect of the Number of Neighboring Vehicles . . . 45
4.4 Simulation Environment . . . 45
4.5 Sensitivity Analysis . . . 48
4.5.1 Sensitivity to Prediction Threshold . . . 48
4.5.2 Sensitivity to Vehicle Density . . . 49
4.5.3 Sensitivity to Network Load . . . 50

Chapter 5: Summary and Conclusions **52**
5.1 Summary . . . 52
5.2 Recommendations and Future Work . . . 53

Bibliography **55**

List of Tables

2.1 Numerology and its Corresponding Spacing and Slot Times. 9

2.2 Summary of C-V2X Requirements 15

2.3 C-V2X Requirements Of Advanced Driving Use Cases 16

4.1 Performance of Evaluated Classifiers 41

List of Figures

2.1 Dynamic Scheduling 11

2.2 C-V2X Uu link 13

2.3 C-V2X PC5 link 14

3.1 Architecture I 24

3.2 Architecture II 26

3.3 ADR error vs speed with fixed $W_y = 10$. **Left**: $W_x = 10$. **Right**: $W_x = 20$ 29

3.4 ADR error vs speed with fixed $W_y = 20$. **Left**: $W_x = 10$. **Right**: $W_x = 20$ 30

3.5 Average ADRs Errors Under Varying Horizons. **Left**: ANN. **Right**: CNN-RNN 31

4.1 Proposed scheme for uplink grants in C-V2X 35

4.2 Flow Chart for proposed C-V2X UL resource allocation scheme . . . 36

4.3 C-V2X Dataset Generation Scheme 37

4.4 Dataset: Layouts of US-101 and I-80 highways used in NGSIM datasets. Both roads are multi-lane free ways with entry and exit ramps. The data is collected through 7 top-down cameras capturing the positions of each vehicles every 100ms. 40

4.5 LSTM model: cross entropy losses under 1-5 s prediction horizons . . 42
4.6 LSTM model: state histories are fed to the model at each time step. The output of the model is the probability of performing maneuver within prediction horizon 43
4.7 Feature Importance In Lane Change Scenario **Left**: Normalized Gradient of Loss Function. **Right**: Speed Histories of Target and Surrounding Vehicles 44
4.8 Normalized Gradient of Loss Function In Turning Scenarios **Left**: Turn Left Scenario **Right**: Turn Right Scenario 45
4.9 Sequence Diagram for Proposed Scheme 46
4.10 Average delay and the corresponding wasted resources 49
4.11 Density vs reliability for 12KB and zero network load 50
4.12 Sensitivity For Increased Background Traffic **Left**: Impact on Latency **Right**: Impact on Reliability 51

Chapter 1

Introduction

1.1 Overview and Motivation

Autonomous vehicles (AVs) are an essential component of future intelligent transportation systems (ITS). AVs promise an increase in driving comfort, greater safety, and improved fuel consumption. These benefits are attributed to AVs ability to leverage vehicle-to-everything (V2X) communication to coordinate driving. Examples of coordinated driving include sharing maneuver intention and receiving updated road maps. Two telecommunication technologies at the forefront of ITS are: cellular-based V2X (C-V2X) and dedicated short-range communication (DSRC). DSRC technology faces a major challenge — its lack of scalability; when the number of connected vehicles increases the DSRC communication link suffers from significant performance degradation [31, 57] due to the contention over the wireless medium. Conversely, C-V2X is more stable [46] because of the controlled scheduling; however, it is questionable if it will support AV advanced use cases. The 3rd Generation Partnership

Project (3GPP) standard defines service requirements for each use case under various degrees of automation (i.e., the degree of human driver involvement in vehicle control) [2]. These requirements vary considerably, for example, use cases such as cooperative lane change requires a latency within 10-25 ms while vehicle platooning requires 10-500 ms latency depending on the supported level of automation. The previous generation of C-V2X (4G LTE) had been proven capable of retaining 100 ms latency while broadcasting to 150 vehicles [46]; the lack of support for advanced use cases at the highest degree of automation is still an obstacle. 5G New Radio (NR) is paving the road to support higher levels of automation by promising 10 ms latency and 99.99% success rate under moderate network conditions. However, at high traffic demand, the NR networks may not be able to support AV requirements for advanced use cases. To the best of our knowledge, there has been very few published works on the feasibility of 5G-based C-V2X in supporting the more demanding advanced AV use cases in a vehicular environment.

1.2 C-V2X Challenges

C-V2X has three challenges: security violation, infrastructure and economic considerations, and performance. Each challenge is discussed further below.

1.2.1 Security

In 2016, a survey conducted by Schmidt et al. [59] found that while the general public accepts the safety benefits of V2X, the loss of privacy, particularly through the revelation of personal data, is seen as unacceptable. Privacy concerns include

data loss to potentially malicious eavesdroppers and commercial entities, and the possession and use of vehicular data by infrastructure providers and governmental authorities. Since the mobile network is designed to incorporate secure authentication and billing through subscription-specific identification, it is uncertain to what extent one's anonymity and privacy can be guaranteed.

1.2.2 Infrastructural and Economic Considerations

In recent years, mobile traffic has experienced incredible growth [40, 41, 29], accompanied by the advent of 5G networks. 5G leverages millimetre waves within a small coverage area, 500 square metres per cell compared to 15 square km in LTE [8]. This size of coverage mandates a dense deployment of 5G base stations. However, economic considerations are still an obstacle to achieving such a goal [51]. For example, operational costs, including upgrading equipment and maintenance of large numbers of base stations, creates an additional expense [50]. It is expected that V2X services will collect a tremendous amount of data. Internet providers charge very high fees for data collection services, making V2X data collection economically unfeasible [50]. These economic considerations have led to an increased interest in device-to-device (D2D) communication. D2D offers an economically feasible solution by removing the need for large scale cell coverage making its operational cost extremely low. C-V2X supports both D2D and cell-based communication, discussed in detail in subsection 2.3.2.

1.2.3 Performance

C-V2X has been shown to scale more stably compared to DSRC [44]; however, congestion control over LTE remains challenging with densely populated cells. Analyses in [63] and [27] confirm that as the number of vehicles in a given cell exceeds a few hundred, beacon delivery rates for unicast Cooperative Awareness Messages (CAMs) drops to unacceptably low levels.

A comprehensive analysis of the performance of LTE C-V2X may be found in [46],The authors report that LTE retains 100 ms latency while broadcasting to 150 vehicles. This is ten times more than the maximum latency allowed for AV advanced use cases such as fully automated cooperative lane change [2], which, at the lowest degree of automation, requires a success rate of no less than 90% in packet delivery with a maximum allowable latency of 25 ms for 300-400 bytes of payload, according to the 3GPP standard [3]. This lowest level of automation typically includes transmitting only the intention of maneuvering. To reach a higher level of automation, a vehicle is expected to transmit additional information, such as, estimated future trajectory and sensory data [2]. To fully support cooperative driving at the highest level of automation, service requirements increase up to 10 ms of latency, for a payload of 12KB at a rate of 99% reliability [39]. These strict requirements could be met with 5G NR, which promises a packet delivery rate up to 99.99%, and latency as low as 10ms [51] at low traffic and low vehicle density use cases. However, to the best of our knowledge, there has been no comprehensive analysis of the performance of NR in a vehicular communication environment at higher packet loads especially when increasing the vehicle density.

1.3 Objective and Contributions

The objective of our work is to study the feasibility of 5G NR in supporting AV advanced use cases and propose solutions to enhance AV performance. We achieved this goal by resolving two bottlenecks present in the current technology: relying on preambles to estimate channel state information (CSI) and excessive delay resulting from the current protocol for resource allocation. Our two contributions are summarized as follows, 1) reduce the delay resulting from preamble transmission and configuration, and 2) reduce the waste of resources in preambles. We detail each contribution below.

1. **Contribution I** We consider the problem of using preambles in estimating CSI in wireless communication systems. Preambles are a signal used in network communications to synchronize the transmission timing between two or more systems. This is done by transmitting a predefined vector of bits by which the receiver can deduce the channel state by comparing the expected signal against the received signal. The drawbacks of this approach are the delay resulting from preamble transmission and configuration, and the resources that are wasted in this preamble since it does not carry actual data. Instead, we propose relying on deep-learning models to predict CSI on the receiver side based on historical data.

   Our approach eliminates the latency overhead and waste of resources resulting from preambles. We evaluated the prediction accuracy of two deep-learning architectures. The first architecture is based on a combination of convolutional and recurrent neural networks (CNN-RNN) and the second architecture is based on multi-layer perceptron (MLP). Both architectures were studied in

a dynamic setting, and we evaluated their sensitivity to vehicle speed, prediction, and history horizons. Interestingly, we have shown that the simpler MLP model performs much better than the more sophisticated CNN-RNN model at a broader range of driving speeds as long as the prediction horizon is smaller than the history horizon in the prediction process.

2. **Contribution II**

   Traditional schemes for uplink resource allocation either have high latency or waste resources. Dynamic resource allocation relies on a round trip of control message exchange to configure the required resources. The delay from this process alone can exceed the maximum acceptable latency required to support advanced AV use cases. High latency could be avoided using semi-persistent scheduling in LTE or configured grants in 5G; however, these approaches lead to a considerable waste of resources. We introduce a novel scheme to enhance uplink resource allocation in C-V2X. The proposed scheme leverages deep-learning predictors using Long Short-Term Memory (LSTM). This prediction module can be installed on the base station (BS) side to predict vehicle maneuver intention using vehicle state histories, including the position, speed, and angle with the road. By predicting the maneuver intention, we allow the BS to proactively schedule uplink grants avoiding the handshake process with minimal waste of resources. Moreover, we prove that C-V2X can support advanced AV use cases under moderate and high traffic loads using our scheme.

1.4 Organization of Thesis

This thesis is organized as follows: Chapter 2 provides a comprehensive literature review of C-V2X, an overview of 5G NR physical layer, scheduling techniques, modes of communication, and use cases. Chapter 3 tackles the problem of CSI estimation, covering the problem formulation, proposed solution, and our results. Chapter 4 considers the problem of excessive latency in dynamic scheduling and resource allocation techniques. The chapter begins with problem formulation, proposed solution, and simulation results. Finally, Chapter 5 concludes and sheds light on future work.

Chapter 2

C-V2X Overview

V2X technologies are regarded as the foundation for future intelligent transportation systems. This chapter explores the C-V2X physical layer, interfaces, scheduling techniques, and use cases.

2.1 Physical Layer

C-V2X can be based on either 4G LTE or 5G NR. In this section, we discuss frame structure, modulation, control messages, and channel state information while focusing on significant improvements provided by NR compared to its predecessor LTE.

2.1.1 Frame structure

NR is similar to LTE, both use a time-frequency resource structure. The resource block (RB) for both generations has 12 consecutive sub-carriers, while the frame and sub-frame durations are 10 ms and 1 ms, respectively. However, NR and LTE differ on the slot level. LTE uses fixed 15 kHz sub-carrier spacing leading to fixed sub-frame

structure comprised of two slots 0.5 ms each. In NR, the sub-carrier spacing, and slot time are flexible and configured according to numerology (see Table 2.1).

μ	$\Delta f = 2^{\mu} * 15[kHz]$	Slot Time (ms)
0	15	1
1	30	0.5
2	60	0.25
3	120	0.125
4	240	0.0625

Table 2.1: Numerology and its Corresponding Spacing and Slot Times.

Moreover, instead of the user equipment (UE) transmitting for the entire duration of a sub-frame. NR introduce mini-slots and multi-slots. A mini-slot can be as short as one orthogonal frequency-division multiplexing (OFDM) symbol and it is particularly useful in scenarios involving small but time critical messages (e.g., C-V2X safety signals). Furthermore, NR V2X allows aggregation to form multi-slots which are suitable for applications that exchange larger than usual packets.

2.1.2 Modulation and Control Channels

NR supports the following channels: 16-QAM, 64-QAM, 256-QAM, 1024-QAM, and QPSK. NR also introduced binary phase-shift keying (BPSK) in UL to enable power efficiency at lower data rates. Moreover, NR introduced a new control channel the physical sidelink feedback channel (PSFCH). PSFCH solves the LTE problem of blind retransmission by having a dedicated channel for source UE to receive delivery acknowledgement messages. Moreover, NR supports the same previous control channels:

- Physical Uplink Control Channel (PUCCH): For uplink control Information (UCI)

- Physical Uplink Shared Channel (PUSCH): For uplink control information (UCI) and application data.
- Physical Downlink Control Channel (PDCCH): For downlink control information (DCI).
- Physical Downlink Shared Channel (PDSCH): For DCI and application data.

2.1.3 Channel Estimation and Reference Signals

5G NR leverages multiple-input multiple-output (MIMO) employed on the BS side. Massive MIMO systems significantly improve the spectrum and efficiency of the NR system making it one of the most promising 5G technologies [28, 52]. A fundamental issue facing the dense deployment of massive MIMO is the overhead of its current technique for acquiring channel estimations (CSI) [34]. CSI characterizes the channel by combining the effects of path loss, scattering, diffraction, and fading. 5G NR relies on preambles; pilot bits of predefined sequence that are decoded on the receiver side to estimate CSI. These pilot bits are signaled through four main reference signals:

- Phase-tracking Reference Signal (PTRS): Used to acquire oscillator phase noise.
- Sounding Reference Signal (SRS): Used to acquire UL-only channel information.
- Channel State Information Reference Signal (CSI-RS): Used to acquire DL-only channel information.
- Demodulation reference signal (DMRS): Used to acquire both UL and DL channel information.

2.2 C-V2X Scheduling

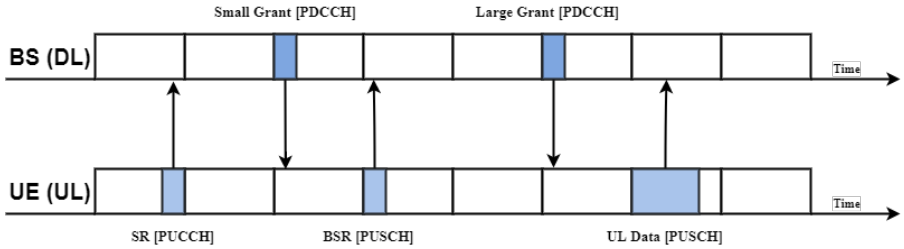

Figure 2.1: Dynamic Scheduling

C-V2X relies on the BS to coordinate resource allocation in PC5 centralized mode (discussed in subsection 2.3.2). To avoid collision, the BS conveys information such as the radio frequency each user is allowed to use in the form of resource grants. The BS can schedule these grants in either a semi-static or dynamic manner.

Semi-static scheduling relies on the BS to periodically grant a fixed number of resources while dynamic resource allocation relies on a handshake protocol in which UE requests resources and waits for the BS assignment (see Figure 2.1). Semi-static scheduling is present in both generations: LTE has semi-persistent scheduling, and 5G NR has configured grants scheduling. Semi-static scheduling mechanism can improve the Quality of Service (QoS) for some applications since it avoids the handshake protocol present in dynamic grants, however, it wastes resources. Moreover, if the number of applications using this approach increases, network congestion could diminish the latency gains [25].

In contrast, in the dynamic grant mechanism the BS should know whether the UE has data to transmit, and how much data the UE has in its buffer to be transmitted. To perform a data transmission efficiently, the BS follows a handshake protocol. First, the BS assigns a small periodic grant for the UE when it first connects to the BS.

This grant, or scheduling request (SR), is typically 1-bit in size given every 10 ms. Due to the small size and long periodicity of SR, it does not cause the network to congest. The UE can use SR messages to signal its need for resources. After the base station successfully decodes the SR, it schedules the UE with an uplink grant.

Using SR alone, the BS cannot know the exact amount of the data that the UE has in the buffer. Thus, typically the BS sends a small UL grant which the UE can use to send a buffer status report (BSR) indicating the approximate amount of data that the UE has in its buffer. Finally, the BS uses BSR information to grant UE the exact number of needed resources. This approach has the advantage of producing minimal waste of UL resources since the BS does not send grants to a UE requested, however, it has the disadvantage of increased latency as the base station needs to wait for SR and BSR before it schedules the right UL grant that satisfies the UE's need. Dynamic grants could follow several sub-paradigms, notably:

- Report scheduling: UEs report information to the BS to assist resource allocation.
- Cross-carrier scheduling: The BS allocate resources on one of the carriers over the other carrier(s).

Choosing between dynamic resource allocation and semi-static is a trade-off between latency and resource consumption. Time critical applications can use semi-static methods to achieve lower latency, while services that require transmission of larger amounts of data use dynamic methods. AV cooperative driving applications are very demanding in terms of latency and payload, for this type of application both dynamic resource allocation and semi-static schemes are problematic.

2.3 V2X Interfaces

C-V2X provides two different interfaces for communication, Uu interface and PC5 interface [64]. PC5 is used in both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). Uu is typically used primarily in V2I although it can theoretically be used in V2V. In the following subsections we discuss these two interfaces in further detail.

2.3.1 Uu Interface

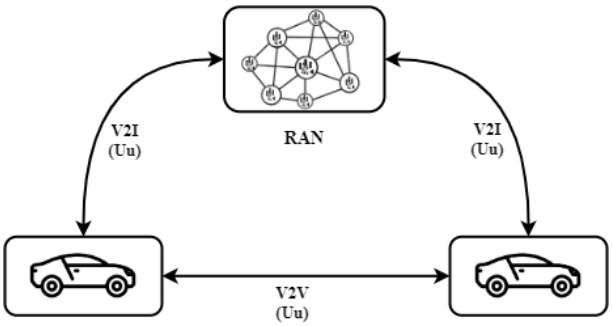

Figure 2.2: C-V2X Uu link

The Uu interface is used for long range communication over a wide area network (WAN) (see Figure 2.2). In this interface, the BS receives a packet on the uplink and relays it to the BS connected to the destination UE which in turn delivers the packets over downlink. In principle this could be used to achieve V2V. However, Uu is expected to be used exclusively in V2I; because of the inherent delays in the network, Uu cannot support high latency critical to safety applications. Uu uses semi-persistent scheduling which will be discussed in 2.2. This technique is only practical for applications that require periodic and similar-sized packets (e.g., software updates and music streams). The main purpose of this interface is to leverage existing LTE

infrastructure and it mainly uses broadcast services for the downlink, where data is transmitted to all/many UE simultaneously.

2.3.2 PC5 Interface

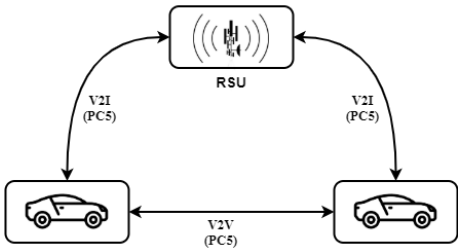

Figure 2.3: C-V2X PC5 link

The PC5 interface, or sidelink, is designed to work in the ITS 5.9GHz band. This interface works with or without network assistance (see Figure 2.3) based on scheduling modes [49]. The centralized scheduling mode, (e.g., mode 1 in 5G NR and mode 4 in LTE), requires network involvement for resource allocation. The decentralized scheduling mode, (e.g., mode 2 in 5G NR and mode 4 in LTE), does not require network involvement. In both modes, a typical packet must include data and control information required to decode the packet (e.g., occupied resources and modulation scheme).

PC5 Centralized Scheduling Mode

This mode is defined for scenarios where BS coverage is available. It is important to note V2X communication happens within the range of only one BS, which makes PC5 working on this mode different from the Uu interface. In this mode, the BS helps in scheduling the allocation of resources for each device. Efficient resource allocation is

essential for PC5 to manage high density traffic and vehicles traveling at high speeds [46]. PC5 mode 3 supports several scheduling techniques, notably semi-persistent and dynamic grant scheduling.

PC5 Decentralized Scheduling Mode

This mode supports device-to-device (D2D) communication connecting the vehicles in an ad-hoc manner and is typically used outside BS coverage. In this mode, resources, (e.g., frequency band), are reserved in a semi-persistent manner using a sensing algorithm. Even with its relatively long sensing period, typically one second, this mode is similar to DSRC in that degradation in performance occurs as the number of connected cars increases [47].

2.4 C-V2X Use Cases

Category	Latency (ms)	Throughput (Mbps)	Reliability (%)
Entertainment Services	500-1000	80	90.0
Traffic Efficiency	100-500	10-45	99.0
Advanced Driving	2-25	50	99.99-99.999

Table 2.2: Summary of C-V2X Requirements

According to the results from [47, 50] both DRSC and C-V2C are capable of supporting a basic set of use cases called Day 1 or safety applications. Most of these applications require the delivery of periodic messages (typically one transmission every 100 ms) and 50-100 ms end-to-end latency. Examples of Day 1 applications are Emergency Electronic Brake Light and Forward Collision Warning, Do Not Pass and Lane Change Warnings [55].

Requirements of some advanced vehicular applications have been studied by the 3GPP in [6] and are summarized in Table 2.2. their requirements are much more stringent than those of basic vehicular applications. Advanced applications use large and variable sized packets and rely on messages that are transmitted aperiodically. In this thesis, we are specifically concerned with advanced and cooperative driving applications, which are shown in further detail in Table 2.3

Scenario	Payload (Bytes)	Latency (ms)	Reliability (%)	Note
Platooning	50-1200	10-25	99.99	Depends on Range and Density
Trajectory Alignment	2000	3	99.999	For Emergency Collision Avoidance
Lane Change	12000	10	99.99	-

Table 2.3: C-V2X Requirements Of Advanced Driving Use Cases

2.5 Literature Review

There is a large body of work studying the performance of C-V2X. Several earlier works compared C-V2X performance to DSRC based technologies [4, 44, 47, 62]. These comparisons address PC5 decentralized mode since DSRC standard, IEEE 802.11bd, does not have a counterpart for C-V2X centralized modes. Existing literature shows that C-V2X offers lower latency and higher resistance to interference when compared to DRSC [4, 47]. However, the performance of LTE C-V2X mode 4 degrades with increased traffic density [47]. With higher traffic density, the frequency re-use distance is decreased leading to higher interference between C-V2X and UE. To the best of our knowledge, there is no similar study to address this issue in 5G NR. Second, several proposals to improve resource allocation in PC5 centralized mode

have been made. A comprehensive survey on sharing resource blocks (RB) based on user clustering can be found in [45].

Hybrid schemes in which DSRC is used to assist C-V2X are proposed in [5, 10]. Authors in [48] proposed AI-enabled predictive resource allocation based on mobility patterns. This proposal is of particular interest to our work. The authors assessed the performance of different machine learning (ML) techniques in predicting mobility traces and leveraged this prediction to optimize hand crafted profit function designed for resource allocation; however, their work lacks in terms of using LTE instead of 5G NR as basis for the simulation settings. Moreover, it did not account for classifying the mobility traces into the corresponding maneuvers. Maneuver prediction model could be leveraged to better allocate network resources, as we prove in this thesis. However, the benefits can vary to a degree based on the performance of the prediction model.

Maneuver classifiers use past motion states of the vehicles as features. Random forest, Bayesian methods, hidden Markov models, and recurrent neural networks (RNNs) have been used to perform this task [58, 60, 14, 36]. Many approaches take into consideration visual cues, such as braking lights, to predict the future motion of the surrounding vehicles [61, 17, 30, 54]. Other works, [35, 58] implicitly learn vehicle interaction from trajectory data of real traffic. A number of combined schemes were proposed in [58, 22, 7].

A comprehensive survey of maneuver-based models can be found in [37, 61]. Finally, the problem of channel estimation was addressed to enhance C-V2X while moving forward from LTE to 5G NR. For example, the works in [26] and [43] proposed least-square error (LS) and minimum-mean-square error (MMSE) estimation

schemes, respectively, to predict the channel state information (CSI). In [16], the authors proposed a maximum likelihood-based channel estimation to predict CSI of macro-cellular OFDM uplinks in a time-variant environment. In [43], Ma et al. utilized a linear MMSE technique to estimate the CSI of individual channels. However, although these methods reduce the network overhead, they are computationally expensive. The 5G mobile communication is expected to use massive MIMO and OFDM to utilize the spectrum ranging from 3 GHz to 300 GHz [11, 18]. Using conventional CSI estimation methods with their large-scale matrix operations incur a formidable challenge for equipment with insufficient computational resources. This added complexity leads to an increase in processing time, and in turn the end-to-end latency, to an intolerable extent for time-critical applications, such as AVs safety applications.

Chapter 3

CSI Prediction

3.1 Problem Formulation

CSI characterizes the channel between the transmitter and receiver where it combines the effects of various sources of signal distortion including path loss, scattering, diffraction, and fading [38]. The combined effect of aforementioned sources of distortion can be modeled on antenna n from antenna m as a complex multiplier $h_{i,j}$

$$h_{i,j}^{t} = \alpha e^{j\theta} \tag{3.1}$$

Considering a MIMO system with K UE on the receiving side and N_B antenna on the BS side, there is $N_K \times N_B$ path between both sides. Assuming a model free from inter-symbol interference, e.g., realized by OFDM. The relationship between transmitted and received signals over the channel can be described by:

$$Y = \mathbf{H_t} * X + Z \tag{3.2}$$

where $X \in R^{N_B \times 1}$ is the data symbol vector, Z is Gaussian noise, and $H_t \in R^{N_K \times N_B}$ is a matrix that represents the channel effect referred to as CSI.

Channel coefficients are highly correlated through space and time. This correlation could be leveraged to predict future CSI from past CSI. This relationship could be modeled as:

$$\mathbf{H}_{f,t} = f(\mathbf{H}_{p,t})$$

Where $\mathbf{H}_{p,t}$ is the sub-matrix containing historical CSI with dimensions $N \times W_x$, and $\mathbf{H}_{f,t}$ is the sub-matrix containing the predicted future CSI with the dimensions $N \times W_y$. Recall that N is the number of paths while W_x and W_y are windows for previous and future time steps, thus defining the history and prediction horizons, respectively. The problem then is reduced into finding the function f that transforms $\mathbf{H}_{p,t}$ into $\mathbf{H}_{f,t}$, for which we can find an approximation using a prediction mode. Throughout this work we focus on two deep learning architectures that could be used to approximate the function f, The first architecture is a new yet simple architecture built mainly on a fully connected multi-layer perceptron (MLP), while the second architecture is based on the CNN-RNN model of [42] after tailoring it to allow predicting CSI in several future time steps, thus becoming suitable for dynamic vehicular environment.

3.2 Data Generation

We used the Matlab Quadriga toolbox [23, 24, 1] to create the dataset needed to study the effect of mobility on the accuracy of CSI prediction. Quadriga is a simulation

tool that builds realistic environments for communication devices. The simulation environment is composed of different segments for rural areas, highways, and the city of Berlin. A "segment" is a Quadriga concept that allows simulating various scenarios in the same experiment by dividing space into different areas, each area has its own scenario and parameters. We will describe the setup of the data generation scenarios in this toolbox, and provide more details on the generated dataset in the sections below.

3.2.1 Data Generation Setup

In our setup, we start by fixing the receiver (representing a cellular BS) in Segment 2 which simulates a crowded downtown neighborhood in the city of Berlin. The transmitter, which is a vehicle in this particular case, starts 6km away in a rural area by establishing a communication link with the receiver. The vehicle speeds up gradually until it reaches Segment 2, which is a highway having a speed limit of 30m/s. It keeps this speed halfway through Segment 2 then starts to decrease it gradually until entering Segment 3 which simulates a less crowded part of the city with speed 20m/s. Again, halfway through this segment 3 the car starts to gradually decrease its speed until it enters Segment 4, the crowded part of Berlin, with speed 15m/s. Finally, the whole process is repeated in reverse with the vehicle increasing its speed gradually until it leaves Berlin to the highway and from the highway to rural areas then it stops.

During this journey the communication link between the receiver and the transmitter is kept alive and a channel coefficient matrix $\mathbf{H}$ is constantly logged at each

time step of 10 ms. In the Quadriga toolbox, the H matrix is 2D matrix of dimensions $N \times T$ where N is the number of signals' paths between the transmitter and the receiver and T is the number of time steps in the simulation scenario. In this form, H could be represented as $[h_1, h_2, ..., h_T]$ where each column vector h_t represent the channel state at a specific time step. Each element in this vector h_t is represents the complex channel gain, $h_{n,t} = \alpha_{n,t} e^{j\theta_{n,t}}$ where $\alpha_{n,t}$ and $\theta_{n,t}$ is the magnitude and phase of the channel response on the n-th path at the t-th time step. It is important to note that N, (i.e., the number of paths), is set in the simulation as the maximum possible number of paths throughout the trajectory, and that all time steps having less paths between the transmitter and the receiver are padded with zeros for the non-existing paths.

3.2.2 The Dataset

The H matrix, described above, represents the channel parameters that we use in this study. To generate the dataset, we run two different sets of simulation scenarios. The first set included multiple runs for the vehicle with variable speeds, while the second set featured multiple runs each imposing a given fixed speed for the vehicle. For the variable speed cases, we use the defined trajectory explained above. For the constant speed cases, we used the same trajectory but have overridden the speed to maintain at a constant value for the entire trajectory. We reiterate this process with different values of the fixed speed in each run, ranging from 40km/h to 200km/h. Though the upper speed value is high compared to the North American standards, it is enabled by the simulator as these speed limits are allowed in European countries, which is where the simulation trajectory is mimicking. For instance, Autobahn, is a highway

road that connects several German cities including Berlin and that allows such speed limits.

To test the sensitivity of the CSI prediction engine to different history and prediction horizons, we create a pair of sub-matrices $\mathbf{H}_{p,t}$ and $\mathbf{H}_{f,t}$ for every time step t, $\forall\ t \in \{1, \ldots, T\}$, which we will refer to as the historical and future sub-matrices at time step t. The historical sub-matrix $\mathbf{H}_{p,t}$ at time step t consists of the column vectors $[h_{t-W_x}, \ldots, h_{t-1}]$, whereas the corresponding future sub-matrix $\mathbf{H}_{f,t}$ consists of the column vectors $[h_{t+1}, \ldots, h_{t+W_y}]$. W_x and W_y are thus hyper parameters to control the length of the history and prediction horizons, respectively. Throughout our work, we have experimented with different combinations of W_x and W_y, to examine their impact on the prediction error.

For training, we employed 80% of the CSI matrices, and their corresponding history and future sub-matrices from the runs featuring variable speeds, as those mimic more closely the real behavior of vehicles. As for validation and testing, we used the other 20% of the variable speed matrices and their sub-matrices. We also used the matrices generated in the different fixed-speed runs in the validation tests only, to explore the sensitivity of the prediction accuracy for each independent speed, when predicted with the trained model.

3.3 Model Architectures

Each of these architectures takes $\mathbf{H}_{p,t}$ as input and outputs its corresponding $\mathbf{H}_{f,t}$. During the training process, the initial weights for both architectures are set as random values between [-1, 1]. These weights are then updated using the *Adam* optimizer

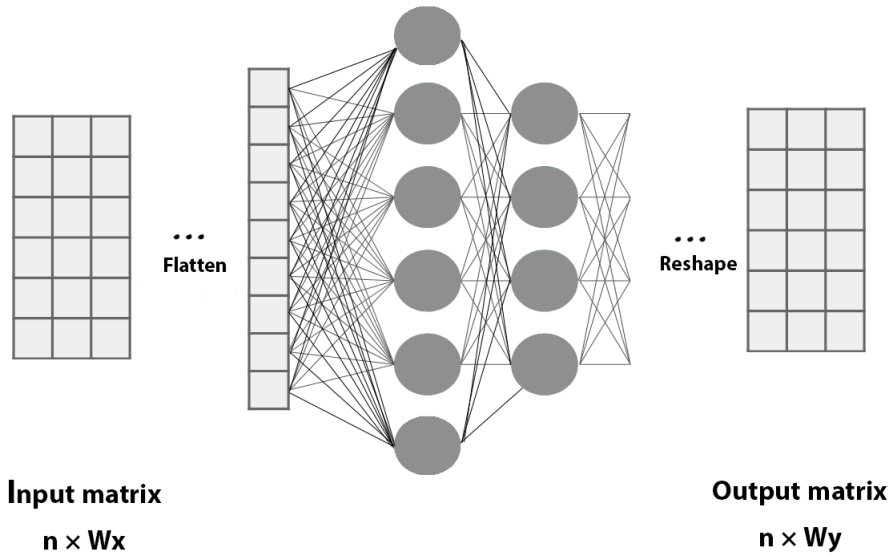

Figure 3.1: Architecture I

[32]. The employed loss function in the training is the mean square error (MSE) metric, defined as:

$$MSE = \sum_{t=1}^{T} \frac{1}{2} \left(y_{f,t} - \widehat{y}_{f,t}\right)^2 \tag{3.3}$$

where $y_{f,t}$ and $\widehat{y}_{f,t}$ are the actual and estimated future CSI vectors, defined as the vectored forms of the actual and predicted future sub-matrix $\mathbf{H}_{f,t}$ and $\widehat{\mathbf{H}}_{f,t}$, respectively; i.e., the vectors including the concatenation of the real and imaginary values of the column vectors of $\mathbf{H}_{f,t}$ and $\widehat{\mathbf{H}}_{f,t}$. Note that these two vectors are of dimensions $2nW_y$

3.3.1 MLP Model

The proposed MLP model consists of three parts: a flatten layer, two trainable dense layers, and finally a reshape layer, as illustrated in Figure 3.1. The flatten layer is

used to convert the input $\mathbf{H}_{p,t}$ into a one-dimensional $y_{p,t}$ vector in a similar manner to the one we used to vectorize of $\mathbf{H}_{f,t}$ into $y_{f,t}$ described above. We will thus refer to $y_{p,t}$, whose length is $2n \times W_x$, as the history CSI vector. Clearly, this vector can be seen as the state vector that contains the CSI features from the time steps before t.

The flatten layer is then followed by two fully connected dense layers. The first dense layer is responsible for prediction. We assume that there exists a real function F that generates the predicted future CSI vector $\widehat{y}_{f,t}$ based on the history CSI vector $\widehat{y}_{p,t}$. Hence, The first dense layer is trained to find an estimated function f where f is defined as $f(y_{p,t}) = \widehat{y}_{f,t}$. We experimented with different settings for the first layer and the best empirical results were observed when using 26 units with $ReLU$ as activation function. The second dense layer is responsible for scaling and preparing the output of the previous layer for reshaping. It consists of $2nW_y$ units to convert the output back into the complex matrix shape of dimensions $n \times W_y$.

It is important to notice that channel coefficients are typically small in value. Hardware limitations lead for these values to be approximated to zero, causing a flat surface problem during the training process. To solve this problem during the training phase, we normalize both the input state vector $y_{p,t}$ and the output $\widehat{y}_{f,t}$ using mini-max feature scaling. Mini-max feature scaling This can be defined by:

$$y_s = \frac{y - min(y)}{max(y) - min(y)} \tag{3.4}$$

Later at the prediction phase, we add a layer for scaling purposes to perform the inverse of this normalization problem, hence the predicted values are scaled back to its normal ranges.

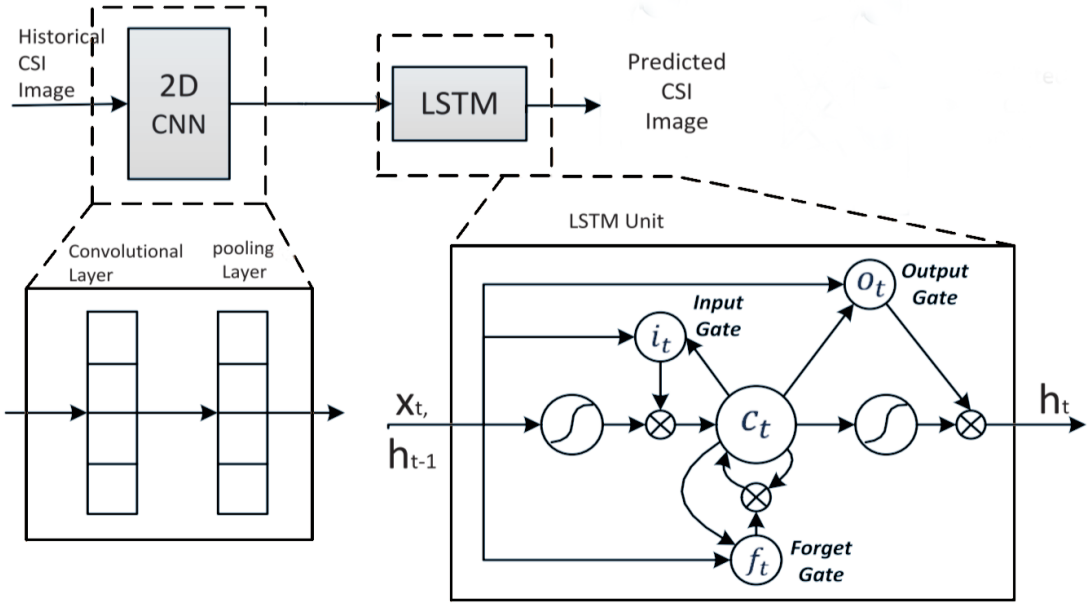

Figure 3.2: Architecture II

3.3.2 CNN-RNN Model

We tailored the CNN-RNN model, originally proposed in [42], to allow CSI prediction in a dynamic environment. The original model was composed of one 2D convolutional layer with kernel size 3×3 and max pooling. Secondly, 1D convolutional layer with kernel size 3×1. Finally, LSTM network followed by a dense layer. In this section, we begin with a summary of modifications applied then we go through a more detailed explanation for the modified model. The main modifications are summarized as follows: First, we replaced the last dense layer with increasing the number of LSTM units. The goal of this modification is to allow training each LSTM unit on predicting the whole CSI vector in a specific time step then use this prediction to predict CSI in the following time step. Secondly, we completely remove the 1D CNN layer. Thirdly, we increase the kernel size of the 2D convolutional layer from 3×3 to 7×7 and change its pooling from max pooling to average pooling. The second and

the third modifications were proven empirically to enhance performance in dynamic environments. The modified CNN-RNN architecture is shown in Figure 3.2

The prediction process in the modified model is the following: each column $h_{f,t}$ and $h_{f,t}$ of the matrices $H_{p,t}$ and $H_{f,t}$ represents a channel state vector at a specific time step. These columns correlate with each other and follow a sequence that depends on both space and time. The intuition behind this architecture is to leverage this correlation to train a neural network on the function F where F is a generator function that takes $h_{f,t-1}$ and S_t as inputs and outputs $h_{f,t}$ where S_t is state vector that represents key features of previous CSI image $H_{p,t}$. However, we need to predict several $h_{f,t}$ columns in the futures, one for each time step. To do so, we repeat the prediction process W_y times predicting one column per repetition. After each successful $h_{f,t}$ prediction, this column is stacked. The end result of this process is the matrix $H_{f,t}$ with the shape $N \times W_y$.

Two critical questions are raised in this process, first, how to find S_t and how to use it to predict $h_{f,t}$. In the following two subsections we discuss this in more detail.

Acquiring state vector S_t

To acquire the state vector S_t we start by feeding previous CSI matrix $H_{p,t}$ to a CNN layer with only one filter of size 7×7. The CNN perform a convolution that could be mathematically expressed as:

$$S_t = H_{p,t} * f$$

where f is the convolution filter with the shape 7

Predicting h_t

Recent advances in deep learning report that recurrent neural networks (RNN) are best suited to learn sequences [20, 21]. We input the state vector S_t to predict the next h then repeat the process stacking the results as explained above. The update of each RNN unit can be briefly summarized as:

$$h_{f,t} = RNN(h_{f,t-1}, S_t, \theta)$$

Where θ is the RNN parameters. There are two main types of RNNs, $LSTM$ and GRU. We experimented with both, and $LSTM$ empirically proved to perform better with long sequences. The LSTM network consists of several units, each of which has an input gate, a forget gate, an output gate, and a memory cell. The mathematical description of the LSTM structure is as follows:

$$\begin{aligned}
i_t &= \sigma(U_{ix}x_t + U_{ih}h_{t-1} + U_{ic}c_{t-1} + b_i) \\
f_t &= \sigma(U_{fx}x_t + U_{fh}h_{t-1} + U_{fc}c_{t-1} + b_f) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \phi(W_{cx}x_t + U_{ch}h_{t-1} + b_c) \\
h_t &= \sigma(U_{ox}x_t + U_{oh}h_{t-1} + U_{oc}c_t + b_o) \\
v_t &= o_t \odot \phi(c_t)
\end{aligned}$$

where i_t, f_t, h_t, c_t, and v_t are input gate, forget gate, output gate, memory cell, and hidden vector respectively, $U_{ix}, U_{ih}\ U_{ic}, U_{fx}, U_{fh}, U_{fc}, W_{ox}, U_{oh}, U_{oc} \in R^{2d}$ are weighted matrices, $b_i, b_f, b_c, b_o \in R^d$, learned during training process, are biases of LSTM, σ is the sigmoid function, $\odot$ stands for element-wise multiplication.

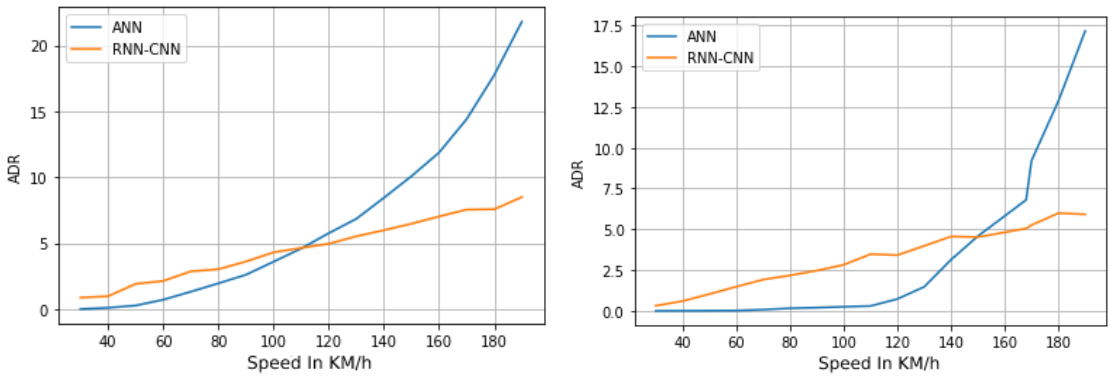

Figure 3.3: ADR error vs speed with fixed $W_y = 10$.
Left: $W_x = 10$. **Right**: $W_x = 20$

3.4 Sensitivity Analysis

We tested both architectures using two experiments. The first experiment studies the mobility effect on the prediction accuracy, the second investigates the effect of history and prediction horizons W_x and W_y.

To assess our models' performance in both experiments we used the average difference ratio (ADR) over all the outputs (of size T) as the performance metric. ADR is defined as:

$$ADR = \frac{1}{T}\sum_{t=1}^{T}\frac{|\widehat{y}_{f,t} - y_{f,t}|}{y_{f,t}} \tag{3.5}$$

3.4.1 Sensitivity to Speed

In the first experiment, we start by fixing the history prediction horizons and test both models against validation data generated from a vehicle moving at each of the fixed speeds we generated data for. We finally take the average of the ADR as a metric of performance for these horizons. We then repeat the above for several horizons to

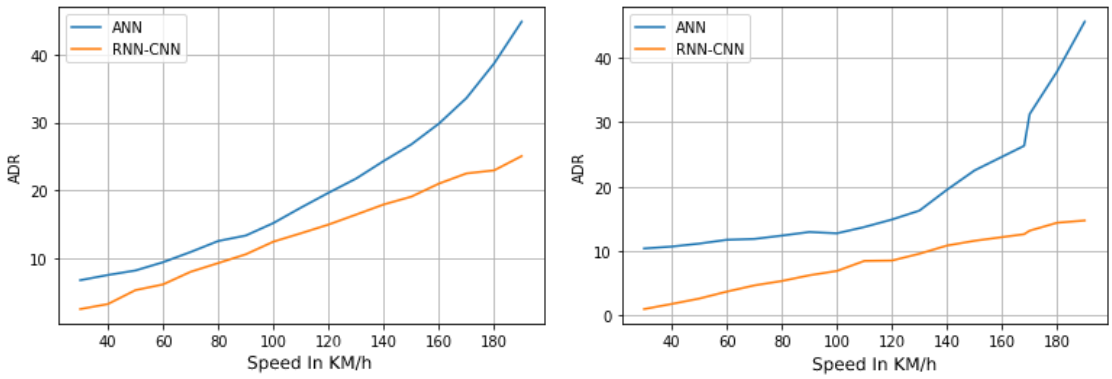

Figure 3.4: ADR error vs speed with fixed $W_y = 20$.
Left: $W_x = 10$. **Right**: $W_x = 20$

compare the impact of horizon selection on the sensitivity to speed change.

The results of this experiment are depicted in Figure 3.3, showing the ADR error results against speed for history horizons $W_x = 10$ and $W_x = 20$, and a prediction horizon $W_y = 10$. Figure 3.4 shows the results for the exact same settings but only for a higher prediction horizon $W_y = 20$.

We can see from both figures that the ADR error performance of both models are highly affected by vehicle speed. In Figure 3.3 we can observe that the MLP model starts off with considerably low ADR error at slow speeds (around 0.3% at 40km/h for both settings) but then the error exponentially increases when the vehicle's speed increases, reaching 10% when $W_x = 10$ and 5% when $W_x = 20$ at 150km/h. This drastic ADR increase with higher W_y for the MLP model is attributed to its failure to capture long-term dependencies.

The CNN-RNN model follows the same trend but with an almost linear increase of the ADR error as the speed increases, as opposed to the exponential increase trend of the MLP model. Though they start at a higher ADR than MLP (e.g., 1% at 40 km/h), then finish with a lower ADR at 150 km/h (6.5% and 4.8% for $W_x = 10$ and

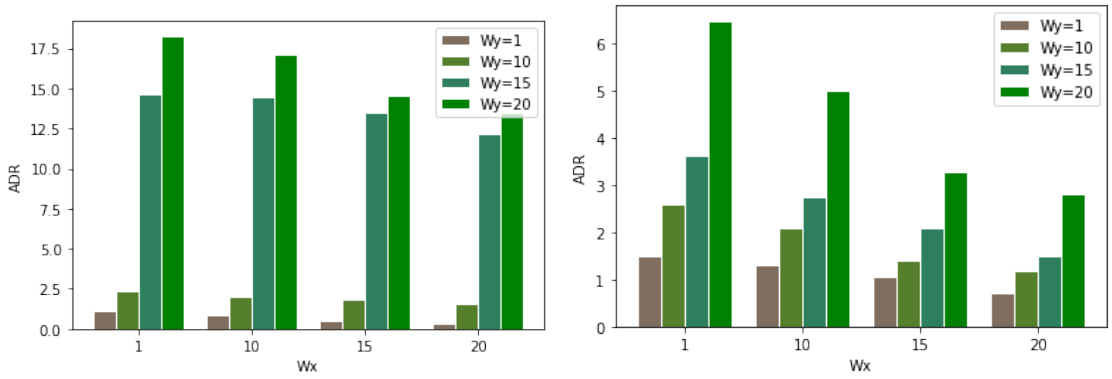

Figure 3.5: Average ADRs Errors Under Varying Horizons.
Left: ANN. **Right**: CNN-RNN

20, respectively). This result is interpreted by the ability of the LSTM memory cells to better capture the long-term dependencies, compared to the MLP model, between columns of $\mathbf{H}_{p,t}$ and $\mathbf{H}_{f,t}$.

Note that these observed trends of increasing ADR with speed are preserved under all history and prediction horizon settings. Yet we can see across all four sub-figures that the absolute values of the ADR errors for both models increase as W_x decrease and as W_y increase, confirming the expectations.

3.4.2 Sensitivity to History and Prediction Horizon

In the second experiment, we test both models against validation data generated from a vehicle moving at 80 km/h and change the chosen history and prediction horizons.

As in the previous section, we take the average ADR as a performance metric for this specific speed.

Figure 3.5 shows the sensitivity performance of both models under different combinations of history and prediction horizons. These results show that CNN-RNN

model is affected when both the history and prediction horizons are changed.

In contrast, MLP architecture is not highly affected by the history horizon W_x, but its performance degrades drastically with increasing prediction horizon W_y. We also observe a large variance in MLP performance under different settings. With high history horizon, and small prediction horizons, e.g. $W_x = 20$ and $W_y = 1$, this model can score ADR as low as 0.7% but with smaller history horizon and larger prediction horizons, (e.g., $W_x = 1$ and $W_y = 20$), the ADR increases to 17.5%.

3.5 Results Discussion

3.5.1 Comparison of the Two Models

We now compare the performances of the MLP and CNN-RNN models from our simulations using different values of speed, history horizon, and prediction horizon. In general, we observe in Figure 3.3 that, for smaller prediction horizons, the MLP model performs better than the CNN-RNN models for most conventional driving speeds (i.e., from 40 to 120 km/h). This superior performance becomes even more visible when the history horizon becomes larger.

On the other hand, for larger prediction horizons, such as the cases depicted in Figure 3.4, the MLP model cannot compete with the CNN-RNN model even at the lowest speeds.

Therefore, network providers must select the suitable scheme based on the employed prediction horizon. Based on these results, it is advised to use the MLP model when vehicles moving at relatively low speed and even up to 120 km/h, as long as as a prediction horizon of up to 100 ms (10 x step duration of 10 ms) is suitable for

the application (e.g., for fast-paced transmissions such as safety messages). On the contrary, the CNN-RNN based model is advised with high speeds (such as the Autobahn highway) or if a high prediction horizon is needed (e.g. for long-term planned transmissions such as maneuver coordination).

3.5.2 Comparison with State of the Art

As noted, the CNN-RNN CSI prediction model provided in [42] is not designed for high mobility environments due to its focus on more quasi-fixed environments. Even when testing their CNN-RNN model in these more controlled and less dynamic environments, an average ADR score of 3.015% was reported. With our added features to account for mobility, our CNN-RNN model still outperforms state of the art, even when tested in harsher and more dynamic environments. Indeed, our CNN-RNN model scores an average ADR of 2.165% across all settings of W_x and W_y for a vehicle travelling with a speed of 80 km/h. We believe this improvement is due to the modifications we applied to the CNN layers, and to better fine tuning the parameters during the training process. Moreover, we prove that, although the average ADR for MLP architecture is higher compared to CNN-RNN, MLP outperforms CNN-RNN when the prediction horizon is small, making it even more suitable for the more stable environments in [42], as long as the application requires fast shorter-term predictions.

Chapter 4

Maneuver Prediction

4.1 Problem Formulation

In current technology, traditional schemes for resource allocation are built upon two paradigms: semi-static and dynamic scheduling. Semi-static scheduling suffers from excessive delay due to its reliance on blind and periodic assignment of resource blocks while dynamic resource allocation relies on a handshake protocol. In this section, we propose a new scheme for UL scheduling (see Figure 4.1 and Figure 4.2). Our system is composed of an autonomous vehicle with an installed onboard UE and BS with an installed prediction model. The BS is responsible for preserving state histories of the vehicle, in particular, its location and speed, for a window of time. Using observed histories of AV states, the BS is able to predict the intention of future maneuvering through its prediction model, hence, it can estimate a future need for UL resources that will be used by the vehicle's onboard UE to then share its intention with the surrounding vehicles during future cooperative path planning. From there, the BS can proactively schedule granting UL resources to this UE, saving the round-trip delay

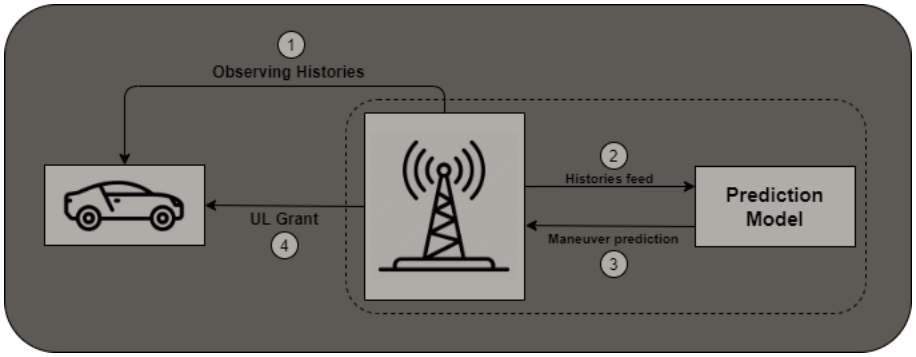

Figure 4.1: Proposed scheme for uplink grants in C-V2X

present in the traditional scheduling scheme. This solution completely eliminates the handshake process replacing it with directly the sending UL data in case of a successful prediction.

To study this system, we need a simulation environment that is able to provide a network simulator and mobility dataset, meaning a dataset that contains vehicle trajectories, speeds and angles with recorded future maneuvers. In the following subsections, we describe the details of acquiring the mobility dataset, and the prediction model, including implementation, correlation of input features with predicted classes, and the effect of including information of surrounding vehicles. We also describe the simulation environment and the link between the mobility dataset and the network simulator. Finally, we present simulation results from sensitivity analysis we performed to study the performance of the proposed scheme for resource allocation based on maneuver prediction.

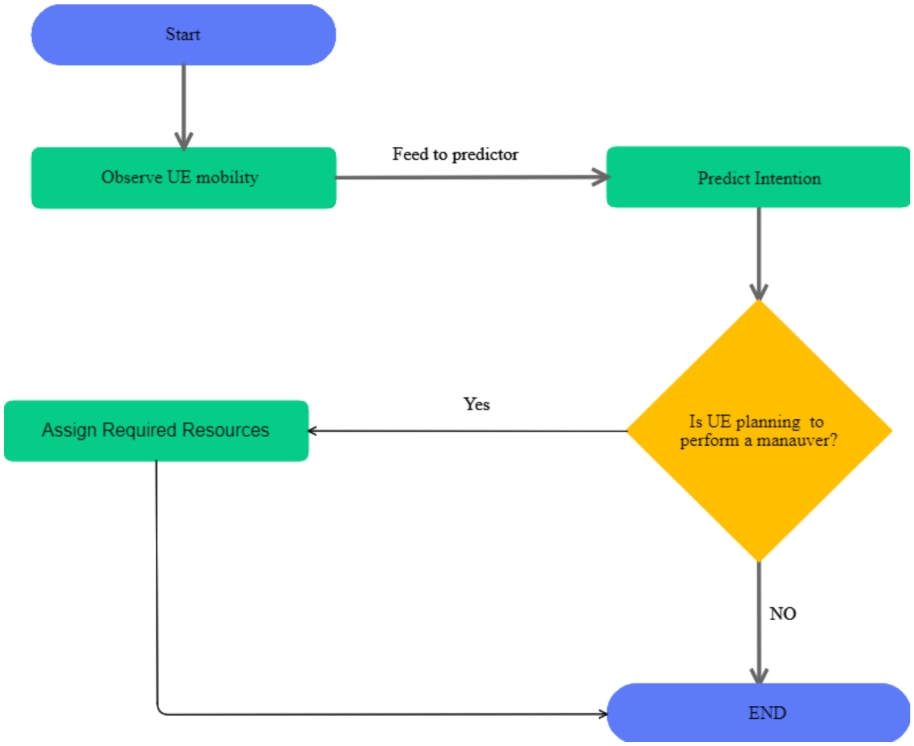

Figure 4.2: Flow Chart for proposed C-V2X UL resource allocation scheme

4.2 Dataset

We provide a generalized scheme for C-V2X dataset generation (see Figure 4.3). This scheme is composed of mobility dataset, network simulator, prediction model, and an interface. The mobility dataset provides realistic state histories, including vehicles trajectories, speed and angle between the vehicle direction and the direction of the road, which can be used by the prediction model to acquire times of maneuvering. This information is processed by the interface and fed to the network simulator, which in turn schedules packet exchange between a dynamic node, i.e., the vehicle, and its surrounding. In the following subsections we discuss our methodology in further detail.

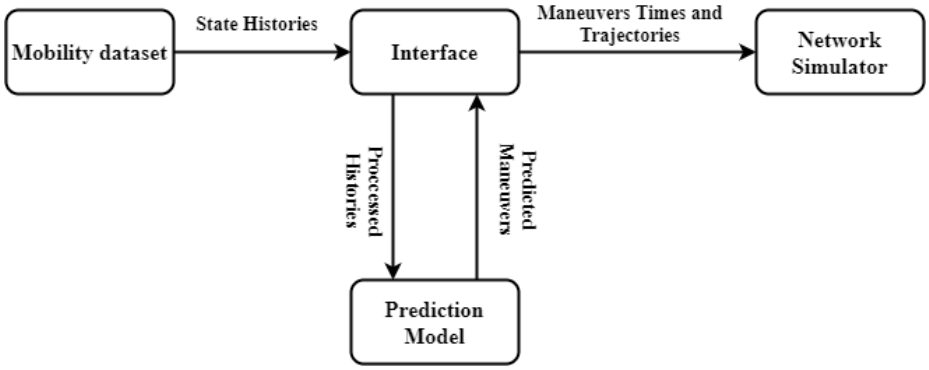

Figure 4.3: C-V2X Dataset Generation Scheme

4.2.1 Mobility Dataset Generation Method

Traffic simulators are classified into macroscopic and microscopic. Macroscopic tools take place on a section-by-section basis providing insight on congestion, traffic density. In contrast, microscopic simulators simulate individual vehicle behaviour and on-lane interaction. Prediction of vehicle maneuver is embedded in mobility patterns for

a particular vehicle; hence, microscopic tools are more appropriate. In our work, we chose to proceed with Simulation of Urban Mobility tool (SUMO) due to its fast execution time and support for OpenStreetMap (OSM) format. OSM is a free, editable map of the world, by integration in SUMO we are able to run our simulations in any desired area of interest.

It is important to note that extracting realistic trajectories from SUMO is not a trivial task. In fact, the mobility dataset resulted from running SUMO with the default configuration is in fact, unusable. In the following subsection we discuss the major challenges faced while generating the mobility dataset from SUMO.

Number of Vehicles and Path Assignment

The Ontario Ministry of Transportation provides statistics regarding the number of vehicles moving from different districts. We can use these statistics to form origin destination (O/D) matrix and feed it to SUMO with the region of interest extracted from OSM. Running the simulation with the default SUMO configuration leads to unbalanced traffic where selected roads are heavily congested while others are nearly empty. The traffic jam is a result of the way SUMO performs path assignment, when applying the Dijkstra algorithm on the map topology graph. As a result, all vehicles choose the route with the shortest path between origins and destinations extracted from O/D matrix. We resort to the Gawron algorithm [19] to solve this issue. This technique initializes the weight of the edges connecting topology graph based on road length and maximum speed allowed on the road, similar to Dijkstra. However, Gawron algorithm iterate the simulation process shifting the path assignment for each vehicle using the traffic density on each road. After a few iterations, the algorithm

reaches dynamic equilibrium and there is no significant difference in path assignments between old and new iterations. This leads to a progressive reduction in traffic jam by forcing the vehicles to choose better routes thus, avoid congestion.

OSM Erroneous Road Information

OSM maps are created by volunteers contributing to the open-source project. Although free and available online, it lacks proper revision leading to the maps containing many errors. These errors may lead to dramatic effect on mobility patterns. The first source of error is wrong restrictions on street segments; some streets are tagged with “straight only“ forcing the vehicle to always continue forward albeit the rule is nonexistent in reality. These erroneous rules do not affect the street layout, however, they can cause vehicles to perform an unneeded long detour instead of performing a turning left or right maneuver. Moreover, SUMO uses ”netCONVERT” package to convert OSM maps to an appropriate format which is also prone to error. For example, some road segments contain multiple values in the source attribute in OSM, netCONVERT is not able to determine the correct value leading to the segment being ignored. This results in disconnected edges in SUMO rendering the entire road unusable. Other roads contain redundant traffic lights that do not exist in reality. We corrected these errors via visual inspection and managed to correct over 500 road segments.

4.2.2 NGSIM

We use the publicly available NGSIM US-101 [12] and I80 [13] datasets for our experiments. Each dataset consists of trajectories of real freeway traffic captured at 10 Hz

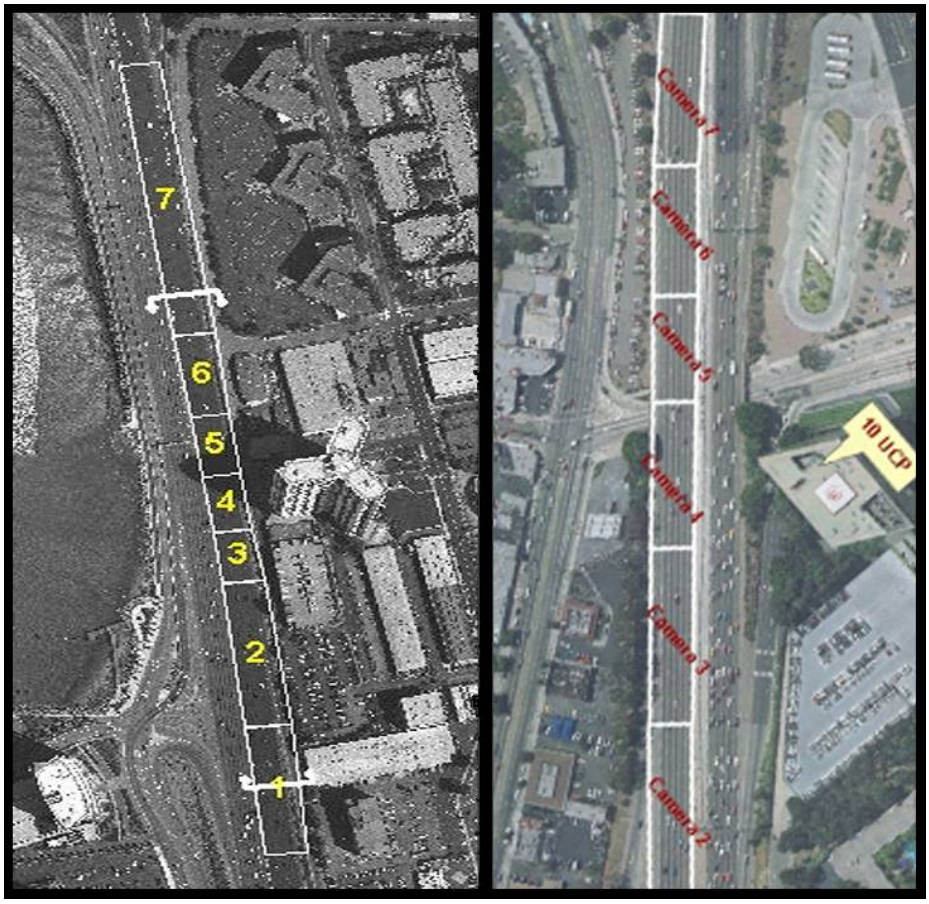

Figure 4.4: Dataset: Layouts of US-101 and I-80 highways used in NGSIM datasets. Both roads are multi-lane free ways with entry and exit ramps. The data is collected through 7 top-down cameras capturing the positions of each vehicles every 100ms.

over a span of 45 minutes. Each dataset consists of 15 min segments of mild, moderate and congested traffic conditions. The dataset provides the coordinates of vehicles projected to a local coordinate system. The total number of trajectories after processing the data is 3121 trajectories including 1184 trajectories involving a lane change. We split the datasets into train and test sets with the test set being one fourth of the data. We exclude vehicles with less than 6 s of saved trajectories. We also down sample each trajectory by a factor of ten to reduce the prediction model complexity

and match ns-3 which updates the locations every 100ms. Moreover, hand-crafted features were added to the data, for example, a feature indicating the angle between the the vehicle and the lane direction, and five features indicating if a lane change is happening within five different prediction horizons.

Finally, since the lane change maneuver involves preparation on the vehicle side, we define a vehicle to be in a lane changing state for ± 2s w.r.t. the actual cross-over.

4.3 Prediction Model

4.3.1 Performance of Different Classifiers

	1s	2s	3s	4s	5s
SVM	82.1%	79.8%	77.4%	74.1%	70.7%
MLP	74.3%	70.9%	66.5%	62.2%	58%
Random Forest	84.3%	81%	78.3%	75.0%	71.6%
LSTM	89.7%	88.3%	87.1%	84.6%	82.0%

Table 4.1: Performance of Evaluated Classifiers

We performed an extensive analysis for different predictors to achieve the highest possible accuracy in predicting the maneuver intention. The performance of different classifiers is summarized in Table 4.1. LSTM came on top with the highest accuracy. Figure 4.5 shows LSTM cross-entropy losses under prediction horizons varying from 1-5 seconds. We believe the inferior performance of other predictors is due to their inability to make use of history information. The maneuver information is encoded within the sequence of events, that is, the relative change in speed and angle compared to previous states. Recurrent neural networks (RNNs) are able to discover relationships between consequent states resulting in an outstanding performance.

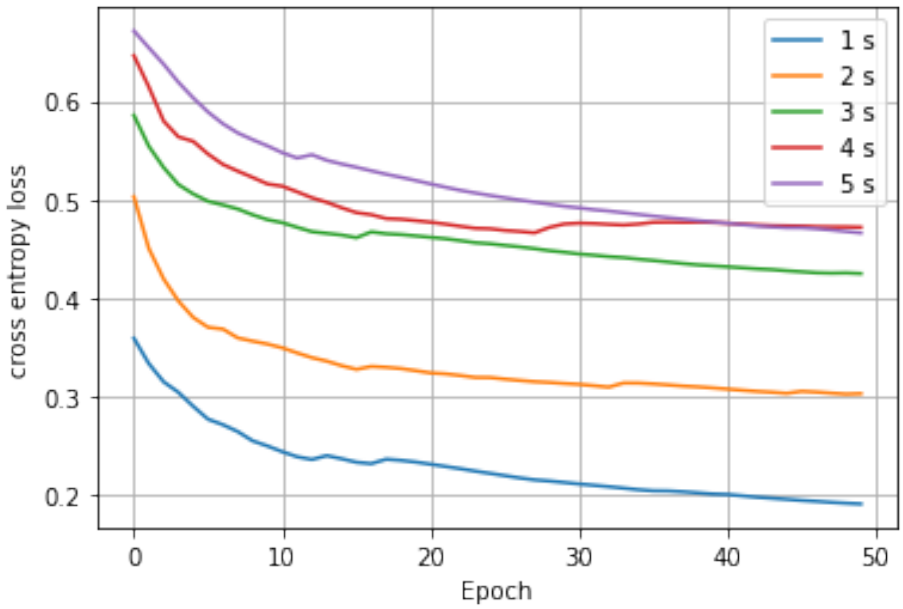

Figure 4.5: LSTM model: cross entropy losses under 1-5 s prediction horizons

4.3.2 LSTM Model

Since the LSTM produces the best prediction performance, in this subsection, we will explain the LSTM implementation.

4.3.3 Implementation

The input to our model is the tensor of track histories

$$X = [x^{t-\tau_h}, ..., x^{t-1}, x^t]$$

where,

$$x^t = [s_0^t, s_1^t, ..., s_n^t]$$

Where τ_h is the length of track histories, n is the number of considered surrounding

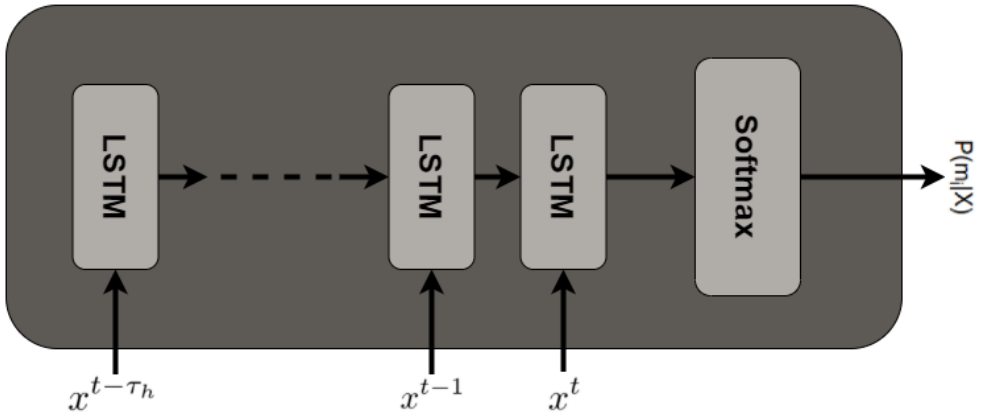

Figure 4.6: LSTM model: state histories are fed to the model at each time step. The output of the model is the probability of performing maneuver within prediction horizon

vehicles, s is the state of the surrounding vehicles, that is, x,y coordinates, speed and the angle between the vehicle direction and the road, at time t.

The output is formulated as a single variable Y expressing the probability of the vehicle performing a lane change within the prediction horizon. Provided with sufficient data to train on, this could be easily extended to 1D vector representing the probabilities of different maneuvers. Moreover, the model is trained to minimize the the sum of cross-entropy losses of the predicted and ground truth of maneuver classes. Cross-entropy loss is defined as:

$$loss = -\sum p(X) log p(X)$$

Finally, use 64 units with leaky ReLU activation function at every time step t of the LSTM. The training was performed using Adam optimizer [33] with a learning parameter of 0.05 and the model was implemented using Keras [9].

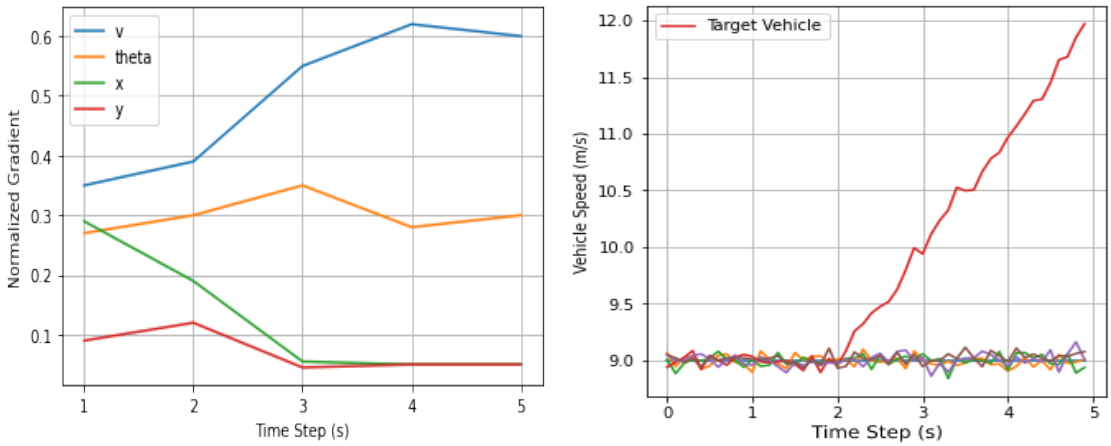

Figure 4.7: Feature Importance In Lane Change Scenario
Left: Normalized Gradient of Loss Function. **Right**: Speed Histories of Target and Surrounding Vehicles

4.3.4 Feature Importance

To assess the feature importance, we separate the data into X folds where each fold contains only classes for a specific maneuver. For each fold, we calculate the normalized gradient of the loss function with respect to the input features at each time step. Interestingly, each maneuver type proved to correlate with specific features. Turn left/right maneuvers correlate with local position (see Figure 4.8). We believe this behaviour is attributed to the network's ability to detect the current lane embedded in local position where vehicles closer to the left of the road are encouraged to turn left and vice versa. In contrast, lane change maneuvers are greatly affected by vehicle speed. Close examination to lane change maneuvers reveal that the model is encouraged to predict lane change if the speed difference between the target vehicle the rest of the traffic is relatively high and vice versa, for example Figure 4.7 shows normalized gradient of the loss function in case of lane change (left) and speed histories for this maneuver. We observe that the target vehicle is accelerating compared to the

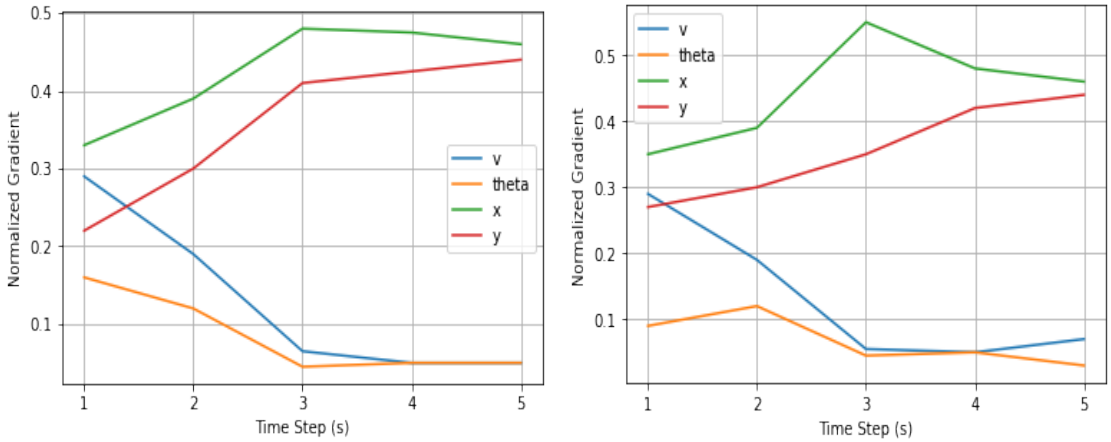

Figure 4.8: Normalized Gradient of Loss Function In Turning Scenarios
Left: Turn Left Scenario **Right**: Turn Right Scenario

surrounding vehicles.

4.3.5 Effect of the Number of Neighboring Vehicles

Our empirical study has shown that the best performance comes with the use of 5 surrounding vehicles. Both adjacent, on the target lane, and leading vehicles are shown to have the greater effect on the performance of the prediction model. Excluding these two vehicles from the model input decreases the accuracy of prediction model by 8.3%.

4.4 Simulation Environment

We used 5G-LENA implementation for the NR [53] to build an environment in which we evaluate the performance of packet exchange between vehicular nodes. 5G-LENA is an open-source for NR simulation, designed as a pluggable module to the ns-3 network simulator [56]. In our environment, vehicles are modeled in ns-3 as a mobile

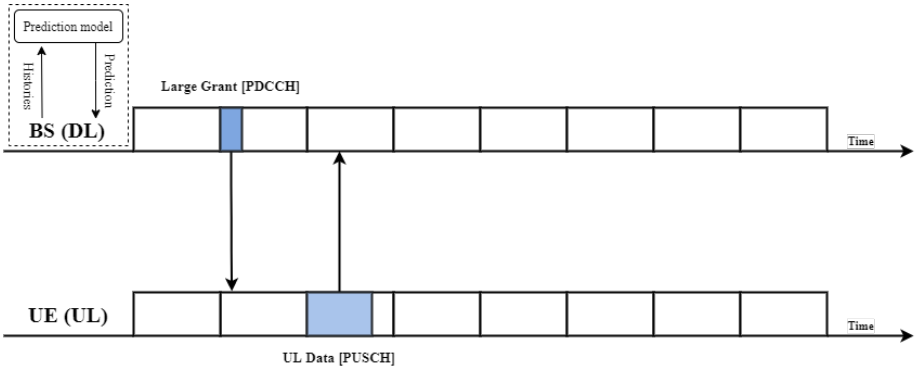

Figure 4.9: Sequence Diagram for Proposed Scheme

nodes communication through a BS. Each vehicle has a UDP-server, and a UDP-client installed. The UDP clients and servers are used to exchange cooperative driving packets which contain information regarding the intention of maneuver. The payload for maneuver message is 12KB. Each vehicle communicates through an antenna installed at a height of 1.5 m, with a transmitting power of 40 dbm. The data rate for each vehicle is 53Mb/s. This rate includes data sharing for both cooperative maneuvers and cooperative perception as specified by the 3GPP standard for cooperative driving in V2X communication in 5G [2].

In our experiment, we use only one BS to forward the maneuver messages between moving vehicles. This BS is centered between the moving vehicles, and it has 64 isotropic antennas installed at a height of 35 m. Moreover, in our environment we only use a single band of 20 MHz at central frequency of 3.5 GHz. Moreover, we use ns-3 Rural Macrocell (RMa) as a propagation loss model to simulate signal propagation in a highway environment. To simulate background load, we create additional nodes, fixed in place and attached to the BS, that are constantly exchanging packets on both uplink and downlink. The packet inter-arrival interval for this background traffic is also drawn from a uniform distribution with the maximum delay 10 ms, and the

packet size is 500 bytes. The number of fixed nodes is a variable to control the background traffic load on the BS during the experiment.

Finally, to compare the performance of the two schemes, we build a scenario in which the locations of mobile nodes are updated every 100 ms based on positions extracted from NGSIM dataset. We schedule ns-3 events that perform maneuver data exchange between a mobile node and its surrounding vehicles. These events are scheduled at the maneuver times also extracted from the aforementioned dataset. Based on this setting the mean maneuver packets delay is collected and and used as a metric to evaluate the ability of the traditional uplink dynamic scheduling scheme to NR the cooperative driving application.

To assess the performance of the proposed scheme, we collect statistics on the latency resulting from uplink grant scheduling then shift the scheduling times of maneuver packet exchange back in time based on the collected statistics and maneuver prediction. The goal is to compensate for the latency resulting from the scheduling process whenever the maneuver is predicted correctly in advance. The required statistics could be collected in ns-3 by logging transmission times of control messages (SR, PUCCH, and PDCCH) on the mac layer of the netDevice installed on mobile nodes. Specifically, the following steps are executed to assess the performance of the proposed scheme:

1. Conduct the aforementioned experiment using a dynamic UL scheduling scheme.
2. For each vehicle: log the times of receiving large UL grants over PDCCH.
3. Outside ns-3: run the predictor on vehicle trajectories
    (a) For false negative predictions, when a vehicle is predicted as not intending

to perform maneuver while it intends to perform a maneuver: schedule the transmission time as t where t is the transmission time as in step 1 left intact.

(b) For false positive predictions, meaning when a vehicle is predicted as intending to perform a maneuver while it does not intend to perform a maneuver: add new transmissions to the ns-3 scheduled maneuver files at time t = t_v where t_v is the time of the last saved coordinate for the vehicle extracted from the mobility dataset.

4. Repeat the experiment with the modified scheduling times.

4.5 Sensitivity Analysis

4.5.1 Sensitivity to Prediction Threshold

Both average delay and waste of resources in the proposed scheme could be controlled by manipulating prediction thresholds α where α is defined as the minimum probability required by the model to predict maneuver intention. Increasing the prediction threshold α results in being more conservative in predicting maneuvers, hence, less waste of resources at the expense of average delay. Figure 4.10 shows the trade-off between waste of resources, defined as the number of times unneeded resources are granted, and average delay with different periods for different values of α in the proposed method. The dynamic scheme performs best in terms of not wasting resources, since it does not suffer from waste, however, it exhibits higher delay compared to our scheme. Increasing the prediction threshold results in the LSTM being more conservative in predicting maneuvers, hence, the number of vehicles benefiting from

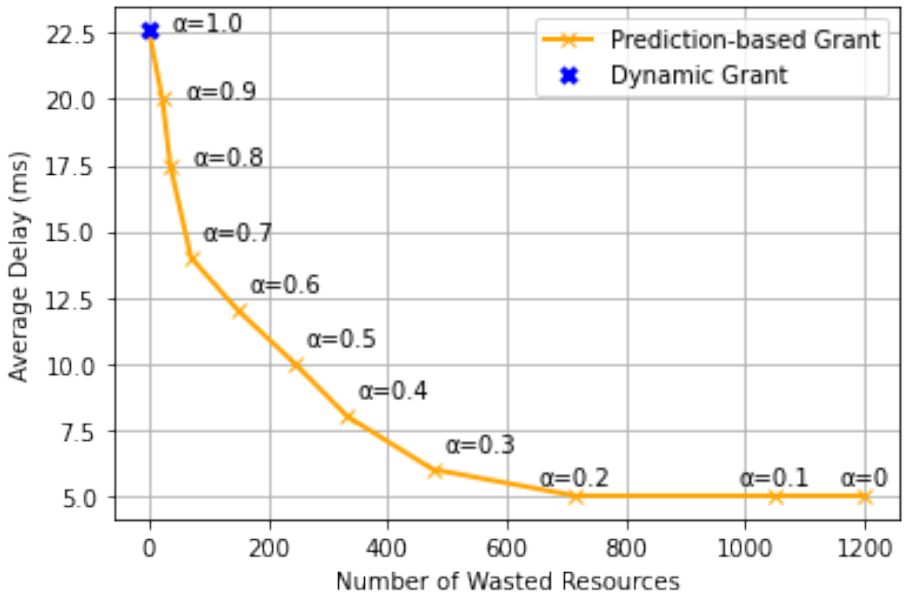

Figure 4.10: Average delay and the corresponding wasted resources

the proposed scheme decreases leading to less waste of resources at the expense of average delay.

4.5.2 Sensitivity to Vehicle Density

Our empirical study has shown that, the density of connected vehicles, defined as the number of neighboring vehicles that acknowledge receiving maneuver intent, has minimal effect on average delay. However, the reliability of package delivery decreases considerably when vehicle density increases. Figure 4.11 shows the density vs reliability for 12KB and zero network load.

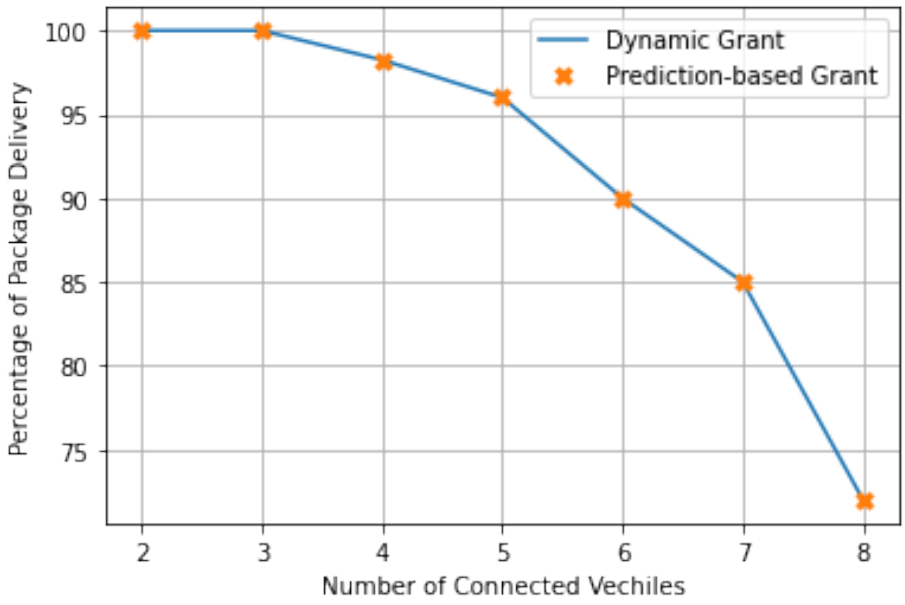

Figure 4.11: Density vs reliability for 12KB and zero network load

4.5.3 Sensitivity to Network Load

Figure 4.12 shows the packet delivery rate and end-to-end latency for both traditional and proposed schemes. Even under relaxed network conditions, the traditional scheme for resource allocation is not able to support the minimum 10 ms delay required by the standard. In contrast, our proposed scheme is able to safely support the standard requirements, however, increased background traffic on the BS side results in a gradual degradation of performance reaching failure at 0.8 GB of data. Moreover, increased background traffic also results in increased packet loss and a degradation of reliability. The number of dropped packets in both schemes is relatively close resulting in almost identical reliability scores. It is important to note that both schemes fail to satisfy the standard reliability requirements under increased network conditions.

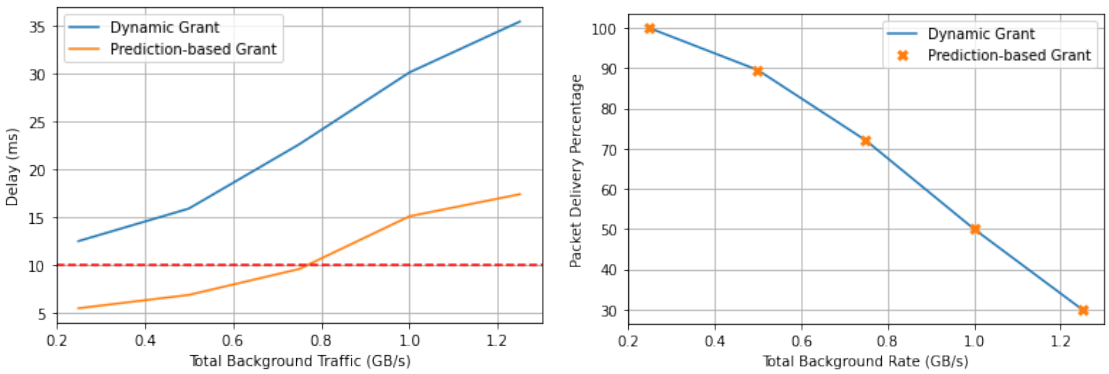

Figure 4.12: Sensitivity For Increased Background Traffic
Left: Impact on Latency **Right**: Impact on Reliability

Chapter 5

Summary and Conclusions

5.1 Summary

In this work, we investigated two bottlenecks present in the current 5G NR standard preventing it from fully supporting AV advanced use cases, specifically, relying on preambles for CSI estimation and the handshake protocol in dynamic resource allocation. For CSI estimation, we resorted to deep learning models to predict channel state information instead of relying on preambles. We analyzed the capabilities and sensitivities of two deep learning models, namely MLP and CNN-RNN networks, in performing uplink CSI prediction for 5G mobile communication channels in vehicular environments. The CNN-RNN model was previously introduced and proven as state-of-the art in [42] but studied only in a fixed indoor environment. We tailored this model to cope with vehicular settings and performed an extensive analysis and comparative study against our novel MLP-based model. Interestingly, our analysis showed that each model has superiority under specific circumstances, and thus each model should be applied according to the surrounding environment and the targeted

application. The MLP-based model can achieve high accuracy with low computational cost as long as the prediction horizon is below 100 ms. On the other hand, the tailored CNN-RNN model should be used at extremely high speeds, or for longer prediction horizons.

Moreover, to solve the bottleneck problem present in the current resource allocation mechanisms, we built a simulation environment , using ns-3 network simulator and 5G-LENA module, to study the feasibility of 5G NR to satisfy AV advanced driving use cases as defined by 3GPP C-V2X standard. Our study has shown that current network capabilities are not able to meet safety constrains required by the standard, hence, we introduced a novel scheme to enhance uplink resource allocation in C-V2X. The proposed scheme leverages deep learning prediction using LSTM. This prediction module should be running on the BS to predict vehicle maneuver intention allowing the BS to proactively schedule uplink grants. We also performed a sensitivity analysis on the LSTM network to study feature importance and the effect of including state histories of surrounding vehicles on prediction accuracy and network performance. The simulation results showed that using our scheme, C-V2X can support advanced driving applications under moderate and high traffic conditions.

5.2 Recommendations and Future Work

A natural extension to this work is creating a unified dataset for CSI, vehicle trajectories in addition to sensory data. The inclusion of sensory data could be achieved by linking the environment we presented in Chapter 4 with an open-source simulator that provides such capabilities, for example CARLA project [15]. This is of particular importance for maneuver prediction model in which the inclusion of visual cues,

(i.e., images from simulated onboard camera), is expected to enhance the prediction accuracy. Moreover, CSI prediction can be extended by building a hybrid model that automatically switches between the two architectures according to the surrounding situation and required application. However, studying the effect of CSI prediction on the network performance is currently unfeasible due to the need for an open-source simulator that could be linked with ns-3.

Bibliography

[1] [online] https://quadriga-channel-model.de/.

[2] 3GPP. Service requirements for enhanced v2x scenarios. Technical specification (ts), 3rd Generation Partnership Project (3GPP), 04 2020. Version 16.2.0.

[3] 3GPP17. 5g v2x; the automotive use-case for 5g.

[4] 5 5GAA. V2x technology benchmark testing. 2018.

[5] Fakhar Abbas and Pingzhi Fan. A hybrid low-latency d2d resource allocation scheme based on cellular v2x networks. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, 2018.

[6] 3GPP auther1. System-level evaluations on sidelink for nr v2x. 2018.

[7] Mohammad Bahram, Constantin Hubmann, Andreas Lawitzky, Michael Aeberhard, and Dirk Wollherr. A combined model- and learning-based framework for interaction-aware maneuver prediction. *IEEE Transactions on Intelligent Transportation Systems*, 17(6):1538–1550, 2016.

[8] Jiaqi Chen, Xiaohu Ge, and Qiang Ni. Coverage and handoff analysis of 5g fractal small cell networks. *IEEE Transactions on Wireless Communications*, 18(2):1263–1276, 2019.

[9] F. Chollet. Keras, 2015.

[10] Hussein Chour, Youssef Nasser, Hassan Artail, Alaa Kachouh, and Ahmed Al-Dubai. Vanet aided d2d discovery: Delay analysis and performance. *IEEE Transactions on Vehicular Technology*, 66(9):8059–8071, 2017.

[11] M. Chowdhury, A. Manolakos, and A. Goldsmith. Multiplexing and diversity gains in noncoherent massive mimo systems. *IEEE Transactions on Wireless Communications*, 16(1):265–277, 2017.

[12] J. Colyar and J. Halkias. Us highway 101 dataset. Technical report, Federal Highway Administration (FHWA), 2007.

[13] J. Colyar and J. Halkias. Us highway 101 dataset. Technical report, Federal Highway Administration (FHWA), 2007.

[14] Nachiket Deo, Akshay Rangesh, and Mohan M. Trivedi. How would surround vehicles move? a unified framework for maneuver classification and motion prediction. *IEEE Transactions on Intelligent Vehicles*, 3(2):129–140, Jun 2018.

[15] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[16] Z. Du, X. Song, J. Cheng, and N. C. Beaulieu. Maximum likelihood based channel estimation for macrocellular ofdm uplinks in dispersive time-varying channels. *IEEE Transactions on Wireless Communications*, 10(1):176–187, 2011.

[17] Jacob Velling Dueholm, Miklas Strøm Kristoffersen, Ravi Kumar Satzoda, Thomas Baltzer Moeslund, and Mohan Manubhai Trivedi. Trajectories and maneuvers of surrounding vehicles with panoramic camera arrays. *IEEE Transactions on Intelligent Vehicles*, 1(2):203–214, 2016.

[18] Z. Gao, L. Dai, D. Mi, Z. Wang, M. A. Imran, and M. Z. Shakir. Mmwave massive-mimo-based wireless backhaul for the 5g ultra-dense network. *IEEE Wireless Communications*, 22(5):13–21, 2015.

[19] C. Gawron. An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. *International Journal of Modern Physics C*, 09:393–407, 1998.

[20] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: continual prediction with lstm. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume 2, pages 850–855 vol.2, 1999.

[21] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, 2017.

[22] Adam Houénou, P. Bonnifait, V. Berge-Cherfaoui, and Wenbing Yao. Vehicle trajectory prediction based on motion model and maneuver recognition. *2013*

IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4363–4369, 2013.

[23] S. Jaeckel, L. Raschkowski, F. Burkhardt, and L. Thiele. Efficient sum-of-sinusoids-based spatial consistency for the 3gpp new-radio channel model. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–7, 2018.

[24] Stephan Jaeckel. *Quasi-deterministic channel modeling and experimental validation in cooperative and massive MIMO deployment topologies*. PhD thesis, Universitätsbibliothek, 2017.

[25] Dajie Jiang, Haiming Wang, Esa Malkamaki, and Esa Tuomaala. Principle and performance of semi-persistent scheduling for voip in lte system. In *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, pages 2861–2864, 2007.

[26] E. Karami. Tracking performance of least squares mimo channel estimation algorithm. *IEEE Transactions on Communications*, 55(11):2201–2209, 2007.

[27] Seiya Kato, Matti Hiltunen, Kaustubh Joshi, and Richard Schlichting. Enabling vehicular safety applications over lte networks. In *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 747–752, 2013.

[28] Imran Khan, Mohammad Haseeb Zafar, Majid Ashraf, and Sunghwan Kim. Computationally efficient channel estimation in 5g massive multiple-input multiple-output systems. *Electronics*, 7(12), 2018.

[29] S. Khan, M. Alam, and M. Fränzle. A hybrid mac scheme for wireless vehicular communication. In *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, pages 889–895, 2017.

[30] Aida Khosroshahi, Eshed Ohn-Bar, and Mohan Manubhai Trivedi. Surround vehicles trajectory analysis with recurrent neural networks. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2267–2272, 2016.

[31] Seungmo Kim and Mehdi Bennis. Spatiotemporal analysis on broadcast performance of dsrc with external interference in 5.9 ghz band, 2019.

[32] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[33] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[34] Petri Komulainen, Antti Tolli, Matti Latva-aho, and Markku Juntti. Channel sounding pilot overhead reduction for tdd multiuser mimo systems. In *2009 IEEE Globecom Workshops*, pages 1–6, 2009.

[35] Eugen Käfer, Christoph Hermes, Christian Wöhler, Helge Ritter, and Franz Kummert. Recognition of situation classes at road intersections. In *2010 IEEE International Conference on Robotics and Automation*, pages 3960–3965, 2010.

[36] Christian Laugier, Igor E. Paromtchik, Mathias Perrollaz, Mao Yong, John-David Yoder, Christopher Tay, Kamel Mekhnacha, and Amaury Nègre. Probabilistic analysis of dynamic scenes and collision risks assessment to improve

driving safety. *IEEE Intelligent Transportation Systems Magazine*, 3(4):4–19, 2011.

[37] Stephanie Lefevre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech Journal*, 1, 07 2014.

[38] Mengyuan Li, Yan Meng, Junyi Liu, Haojin Zhu, Xiaohui Liang, Yao Liu, and Na Ruan. When csi meets public wifi: Inferring your mobile phone password via wifi signals. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1068–1079, 2016.

[39] S. Lien, D. Deng, C. Lin, H. Tsai, T. Chen, C. Guo, and S. Cheng. 3gpp nr sidelink transmissions toward 5g v2x. *IEEE Access*, 8:35368–35382, 2020.

[40] C. Luo, G. Min, F. R. Yu, M. Chen, L. T. Yang, and V. C. M. Leung. Energy-efficient distributed relay and power control in cognitive radio cooperative communications. *IEEE Journal on Selected Areas in Communications*, 31(11):2442–2452, 2013.

[41] C. Luo, G. Min, F. R. Yu, Y. Zhang, L. T. Yang, and V. C. M. Leung. Joint relay scheduling, channel access, and power allocation for green cognitive radio communications. *IEEE Journal on Selected Areas in Communications*, 33(5):922–932, 2015.

[42] Changqing Luo, Jinlong Ji, Qianlong Wang, Xuhui Chen, and Pan Li. Channel state information prediction for 5g wireless communications: A deep learning

approach. *IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING*, 2018.

[43] J. Ma, S. Zhang, H. Li, N. Zhao, and A. Nallanathan. Iterative lmmse individual channel estimation over relay networks with multiple antennas. *IEEE Transactions on Vehicular Technology*, 67(1):423–435, 2018.

[44] Zachary MacHardy, Ashiq Khan, Kazuaki Obana, and Shigeru Iwashina. V2x access technologies: Regulation, research, and remaining challenges. *IEEE Communications Surveys Tutorials*, 20(3):1858–1877, 2018.

[45] A. Masmoudi, K. Mnif, and F. Zarai. A survey on radio resource allocation for v2x communication. *Wirel. Commun. Mob. Comput.*, 2019:2430656:1–2430656:12, 2019.

[46] Z. H. Mir and F. Filali. Lte and ieee 802.11 p for vehicular networking: a performance evaluation. 2019.

[47] Rafael Molina-Masegosa and Javier Gozalvez. Lte-v for sidelink 5g v2x vehicular communications: A new 5g technology for short-range vehicle-to-everything communications. *IEEE Vehicular Technology Magazine*, 12(4):30–39, 2017.

[48] Umer Rehman Mughal, Manzoor Ahmed Khan, Azam Beg, and Ghulam Qadir Mughal. Ai enabled resource allocation in future mobile networks. In *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6, 2020.

[49] Amr Nabil, Komalbir Kaur, Carl Dietrich, and Vuk Marojevic. Performance analysis of sensing-based semi-persistent scheduling in c-v2x networks. pages 1–5, 08 2018.

[50] Mohammed Nadeem Hangar, Qasim Ahmed, Fahd Khan, and Maryam Hafeez. A survey of autonomous vehicles: Enabling communication technologies and challenges. *Sensors*, 21:706, 01 2021.

[51] Arvind Narayanan, Eman Ramadan, Jason Carpenter, Qingxu Liu, Yu Liu, Feng Qian, and Zhi-Li Zhang. A first look at commercial 5g performance on smartphones. In *Proceedings of The Web Conference 2020*, WWW '20, page 894–905, New York, NY, USA, 2020. Association for Computing Machinery.

[52] Sinh Le Hong Nguyen and Ali Ghrayeb. Compressive sensing-based channel estimation for massive multiuser mimo systems. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2890–2895, 2013.

[53] Natale Patriciello, Sandra Lagen, Biljana Bojovic, and Lorenza Giupponi. An e2e simulator for 5g nr networks, 2019.

[54] Derek J. Phillips, Tim A. Wheeler, and Mykel J. Kochenderfer. Generalizable intention prediction of human drivers at intersections. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1665–1670, 2017.

[55] ConVeX Project. convex-project.de.

[56] George F. Riley and Thomas R. Henderson. *The ns-3 Network Simulator*, pages 15–34. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[57] Md Saifuddin, Mahdi Zaman, Behrad Toghi, Yaser P Fallah, and Jayanthi Rao. Performance analysis of cellular-v2x with adaptive and selective power control, 2020.

[58] Julian Schlechtriemen, Florian Wirthmueller, Andreas Wedel, Gabi Breuel, and Klaus-Dieter Kuhnert. When will it change the lane? a probabilistic regression approach for rarely occurring events. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1373–1379, 2015.

[59] T. Schmidt, R. Philipsen, P. Themann, and M. Ziefle. Public perception of v2x-technology - evaluation of general advantages, disadvantages and reasons for data sharing with connected vehicles. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1344–1349, 2016.

[60] M. Schreier, Volker Willert, and J. Adamy. Bayesian, maneuver-based, long-term trajectory prediction and criticality assessment for driver assistance systems. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 334–341, 2014.

[61] Sayanan Sivaraman and Mohan Manubhai Trivedi. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1773–1795, 2013.

[62] Stefania Sesia Valerian Mannoni, Vincent Berg and Eric Perraud. A comparison of the v2x communication systems: Its-g5 and c-v2x. 2019.

[63] A. Vinel. 3gpp lte versus ieee 802.11 p/wave: which technology is able to support cooperative vehicular safety applications? 2019.

[64] J; Weber, R.; Misener. Cellular v2x for safety and cooperative driving of intelligent transport systems. 2018.