DELAY PREDICTION USING MACHINE LEARNING ALGORITHMS FOR CONNECTED AUTONOMOUS TRAFFIC FLOW IN UNINTERRUPTED FACILITIES

by
Kaan Aytekin
B.S., Civil Engineering, Boğaziçi University, 2021

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Civil Engineering
Boğaziçi University
2021

ACKNOWLEDGEMENTS

This thesis is supported by Bogazici University Research Fund (BAP) with project number 15387 and project code 19A04R2 and the Scientific and Technological Research Council of Turkey (TUBITAK 1001) under grant number 120M574.

I would like to express my gratitude and utmost respect to my thesis advisor Assoc. Prof. Ilgın Gökaşar for her guidance and support. This thesis could not be completed without her support.

I would like to thank my family, especially my sister and housemate Burcu Aytekin, for their moral support and patience throughout the writing process.

Finally, I would like to thank my friends and peers Kadircan Çakmak, Sarp Semih Özkan, Rahmi Şahin, Alperen Timuroğulları & Ali Atilla Arısoy for their support on various parts of this thesis.

ABSTRACT

DELAY PREDICTION USING MACHINE LEARNING ALGORITHMS FOR CONNECTED AUTONOMOUS TRAFFIC FLOW IN UNINTERRUPTED FACILITIES

Road incidents and breakdowns in freeways create excess delays in road sections for a duration of time. Amount of this delay plays an important role in the route planning of the people on the road. In this thesis, we propose a delay prediction methodology using machine learning techniques on feature engineered detector data. Our method uses incident characteristics, lagged detector data, adjacent detector data & lagged adjacent detector data as features. Created features are selected using mutual information criteria, correlation analysis and regularized & standard random forest feature importance values. The final model successfully predicts the delay for next timestep with mean squared error of 224.89 for training and 247.77 for testing data sets. Model performance further improves for the simulation conditions with less uncertainty such as incidents with short duration, accidents on right or left lane and detectors further away from the incident location.

ÖZET

KESİNTİSİZ AKIMLARDA BAĞLANTILI OTONOM TRAFİK AKIŞI İÇİN MAKİNE ÖĞRENİMİ ALGORİTMALARINI KULLANARAK GECİKME TAHMİNİ

Otoyollardaki kazalar ve arızalar, bir süre boyunca yol bölümlerinde ek gecikmelere sebep olmaktadır. Bu gecikmenin miktarı, yoldaki insanların rota planlamasında önemli rol oynamaktadır. Bu tezde, özellik mühendisliği ile zenginleştirilmiş dedektör verileri üzerinde makine öğrenim teknikleri kullanarak gecikme tahmini yapan bir metodoloji öneriyoruz. Metodumuz özellik olarak kaza karakteristiklerini, dedektör verilerinin gecikmeli değerlerini, bitişik dedektör verilerini & bitişik dedektörlerin gecikmeli verilerini kullanmaktadır. Oluşturulan özellikler, karşılıklı bilgi kriterleri, korelasyon analizi ve düzenlileştirilmiş & standart rastgele orman özellik önemi değerleri kullanılarak seçilir. Sonuçta elde edilen model bir sonraki zaman dilimi için gecikmeyi eğitim setinde 224.89 ve test setinde 247.77 ortalama kare hata ile başarılı bir şekilde tahmin etmektedir. Model performansı kısa süreli kazalar, sağ veya sol şeritta kazalar ve kaza yerine uzak yol bölümleri gibi belirsizliğin daha düşük olduğu simülasyon koşullarında daha da iyi sonuç göstermektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES viii
LIST OF TABLES xiv
LIST OF ACRONYMS/ABBREVIATIONS xv
1. INTRODUCTION & MOTIVATION 1
2. LITERATURE REVIEW 4
2.1. Delay Studies in The Literature 4
2.2. Delay Studies with Connected/Autonomous Vehicles 6
3. THEORY 13
3.1. Linear Regression 13
3.2. Bayesian Ridge Regression 17
3.3. Lasso Regression 22
3.4. Random Forest 23
3.5. XGBoost 26
4. Data Generation Process 28
5. METHODOLOGY 32
5.1. Feature Engineering 32
5.1.1. Target 33
5.1.2. Lagged Detector Features 34
5.1.3. Adjacent Detector Features 34
5.1.4. Incident Related Features 35
5.2. Simulation Based Data Splitting 35
5.3. Feature Selection Process 38
5.3.1. Correlation Analysis 38
5.3.2. Mutual Information Criterion 39
5.3.3. Regularized & Standard Random Forest Feature Importance . . 41

5.3.4. Feature Selection 42
5.4. Considered Models 46
5.5. Cross Validation, Hyperparameter Tuning & Model Selection 47
5.5.1. Hyperparameter Space of the Models 47
5.5.2. Cross Validation & Model Selection 48
6. Results 52
6.1. Model Results 52
6.2. Discussions 56
7. Conclusion 69
REFERENCES 71
APPENDIX A: PERMUTATION FEATURE IMPORTANCE 77
APPENDIX B: DELAY PLOTS FOR DIFFERENT CONNECTED AUTONOMOUS VEHICLE LEVELS 80
APPENDIX C: SPEED HEATMAP FOR DIFFERENT ACCIDENT LANES & CONNECTED AUTONOMOUS VEHICLE LEVELS 83

LIST OF FIGURES

Figure 3.1. Deviations from regression line. 15

Figure 3.2. Cost function contours and regularization constraints. 18

Figure 4.1. Illustration of the road network. 28

Figure 4.2. Delay Distribution for a Simulation Run with No Incident. 30

Figure 4.3. Delay vs. Timestep Plot of Detectors 9, 10, 11 & 12 for an Incident on Section 11. 31

Figure 5.1. Modelling Process Flowchart. 32

Figure 5.2. Simulation Based Data Splitting Process. 37

Figure 5.3. Correlation Coefficient For Different Relationships. 39

Figure 5.4. XOR Problem. 40

Figure 5.5. 5 Repeated 5-Fold Cross Validation Pseudocode. 50

Figure 6.1. Permutation Feature Importance for Selected Model. 53

Figure 6.2. Shap Values Random Forest Model. 54

Figure 6.3. Shap Values for CAV Ratio Feature of Random Forest Model. . . 55

Figure 6.4. Shap Dependence Plot for CAV Ratio Feature of Random Forest Model. 55

Figure 6.5. Partial Dependence Plot for CAV Ratio Feature of Random Forest Model. 56

Figure 6.6. Delay Prediction Error Plot for Different Incident Lanes. 57

Figure 6.7. Delay Plot When Incident Is On Lane 1 (Rightmost Lane). 58

Figure 6.8. Delay Plot When Incident Is On Lane 2 (Middle Lane). 58

Figure 6.9. Delay Plot When Incident Is On Lane 3 (Leftmost Lane). 59

Figure 6.10. Delay Plot When There Is No Incident. 62

Figure 6.11. Delay Plot When There Is An Incident. 63

Figure 6.12. Delay Prediction Error Plot for No Accident & Accident Scenarios. 63

Figure 6.13. Delay Plot When Connected Autonomous Vehicle Ratio Is 0%. . . 64

Figure 6.14. Delay Plot When Connected Autonomous Vehicle Ratio Is 80%. . 65

Figure 6.15. Delay Prediction Error Plot for Different CAV Ratio Levels. . . . 65

Figure 6.16. Delay Plot When Incident Duration Is Short (607 sec). 66

Figure 6.17. Delay Plot When Incident Duration Is Long (1492 sec). 66

Figure 6.18. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 20%. 67

Figure 6.19. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 40%. 68

Figure B.1. Delay Plot When Connected Autonomous Vehicle Ratio Is 0%. . . 80

Figure B.2. Delay Plot When Connected Autonomous Vehicle Ratio Is 20%. . 81

Figure B.3. Delay Plot When Connected Autonomous Vehicle Ratio Is 40%. . 81

Figure B.4. Delay Plot When Connected Autonomous Vehicle Ratio Is 60%. . 82

Figure B.5. Delay Plot When Connected Autonomous Vehicle Ratio Is 80%. . 82

Figure C.1. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 0%. 83

Figure C.2. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 10%. 84

Figure C.3. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 20%. 85

Figure C.4. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 30%. 86

Figure C.5. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 40%. 87

Figure C.6. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 50%. 88

Figure C.7. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 60%. 89

Figure C.8. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 70%. 90

Figure C.9. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 80%. 91

Figure C.10. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 0%. 92

Figure C.11. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 10%. 93

Figure C.12. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 20%. 94

Figure C.13. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 30%. 95

Figure C.14. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 40%. 96

Figure C.15. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 50%. 97

Figure C.16. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 60%. 98

Figure C.17. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 70%. 99

Figure C.18. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 80%. 100

Figure C.19. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 0%. 101

Figure C.20. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 10%. 102

Figure C.21. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 20%. 103

Figure C.22. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 30%. 104

Figure C.23. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 40%. 105

Figure C.24. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 50%. 106

Figure C.25. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 60%. 107

Figure C.26. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 70%. 108

Figure C.27. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 80%. 109

LIST OF TABLES

Table 2.1 CAV Papers in Literature. 8

Table 4.1. Simulation Parameters. 28

Table 4.2. Simulation Configurations. 29

Table 4.3. Characteristics of the vehicle types. 29

Table 5.1. Original & Lagged Features. 34

Table 5.2. Original & Adjacent Features. 35

Table 5.3 Relevance Ranks for Selected Features. 42

Table 5.4. Model Hyperparameters and Search Spaces. 49

Table 5.5. Cross Validation Results. 51

Table 5.6. Hyperparameters for Random Forest Regressor. 51

Table 6.1. Average Detector Metrics for Different Simulation Configurations . 60

Table 6.2. Detector Metrics on Incident Local Area for Different Connected Autonomous Vehicle Ratio Levels. 61

Table A.1 Permutation Feature Importance Values. 77

LIST OF ACRONYMS/ABBREVIATIONS

CAV	Connected Autonomous Vehicle
FI	Feature Importance
H	Hour
I.I.D.	Identically and Independently Distributed
Km	Kilometers
Kmph	Kilometers per Hour
MI	Mutual Information
MSE	Mean Squared Error
RF	Random Forest
RRF	Regularized Random Forest
S	Seconds
SSE	Sum of Squared Errors
Veh	Vehicle

1. INTRODUCTION & MOTIVATION

Road incidents and breakdowns inevitably happen on busy roads. As a consequence, upstream and downstream drivers experience excess travel time delay [1, 2]. Experienced delay depends on various things such as road capacity, traffic volume, incident lane, distance to the incident and so on [3,4]. In order to understand the characteristics of experienced delay, in this thesis, various incident scenarios with different configurations are simulated.

Level and duration of traffic delay plays an important role in the vehicle route planning decisions of the drivers due to the purpose of decreasing the time spent in traffic [5]. Increased traffic delays stimulate an increase in the travel times, inefficiency, wasted fuel and commuter frustration [5,6]. Therefore, there is a need for the modelling of an accurate traffic delay estimation algorithm.

In literature, many studies regarding traffic delay estimation are present. In several studies, Webster model is modified to be used in delay estimation [7, 8]. In a different study, a recently developed queuing based approach is used to estimate traffic delay [9]. The proposed model by this thesis is different from the majority of the studies present in the literature, since machine learning and feature engineering techniques are used.

In this study, the proposed model aims to predict travel delays using feature engineering and machine learning techniques, which can be used in various fields such as route planning applications. Purpose of such applications is to inform passengers according to the travel times of specific road segments [10], and divert the traffic from the region, which is very beneficial in terms of reducing the wasted fuel, inefficiency, and travel times.

With the recent advancements in connected autonomous vehicle (CAV) technology, implementation of these vehicles into traffic flow is not far into the future [11]. There are various studies in the literature investigating the potential benefits provided from the utilization of CAVs in real traffic. For instance, simulation studies on the subject show an increase in safety and traffic efficiency aspects [11, 12]. However, it is also a known fact that a complete transition to 100% CAV penetration rate in traffic flows is not possible. Therefore, initially, a mixed traffic of human-driven vehicles and CAVs will be at hand.

Utilization of CAVs in real traffic flow and sustaining a mixed traffic of human-driven vehicles and CAVs, great benefits to delay estimation and reduction can be provided. In literature there are many studies, which are related to the effects of CAVs on traffic delays. In a study, the incorporation of human-driven vehicles and CAVs in distributed traffic signal control methodology [13]. Simulations are done on a real-life network. Results indicate 50% to 61% reduction in average traffic delay at 40% penetration rate of CAVs. In a different study, the effects of mixed traffic condition at signalized intersections are investigated [14]. Utilization of CAVs in the traffic flow and using the position and speed data of the individual vehicles make it possible to adapt the traffic signal, which shows up to 60% decrease in traffic delays with CAV penetration rate between 0% and 60%. These studies show that, implementation of CAVs into the traffic flow is highly beneficial in means of reducing traffic delays. One way that the approach of this study is different from the other studies in the literature is that delay estimation models are used. If the traffic conditions increasing delays are estimated before, CAVs can operate more efficiently and effectively due to the preparedness to the situation.

The goal of this study is to propose a delay prediction model using machine learning on feature engineered detector data to be able to present an accurate traffic delay estimation method. This goal is achieved by the following objectives consisting of the utilization of SUMO simulation software and machine learning tools.

Additionally, this study fills a gap in the literature by gathering CAVs, machine learning and feature engineering together in order to provide an accurate delay estimation method.

2. LITERATURE REVIEW

Transportation is one of the indispensable parts of daily life. While many means of transportation make life easier for people to go from one place to another, they also cause some problems and can be easily affected by environmental factors. Transportation can become a big problem especially in the city and during the rush hour, where the traffic is heavy, because heavy traffic can have delays. These delays pose a big problem for the people involved in transportation, which can lead them to different alternative transportation means or spend a large part of the day in transportation. While there are many studies on these delays in the literature, there are some studies on why these delays occur and how they can be reduced or even prevented. There may be delays due to highway tolls, police control, accidents and the excess demand for transportation at the same time intervals. Different suggestions are offered to deal with these delays. In this study, delay estimations for Connected Autonomous Traffic Flow in Uninterrupted Facilities will be made using machine learning algorithms. In order to show how this study can contribute to the literature and how it can be different from other studies in the literature, studies in the literature will be examined under two headings as Delay Studies in The Literature and Delay Studies with Connected/Autonomous Vehicles.

2.1. Delay Studies in The Literature

No matter how good the traffic management is, it is very difficult to prevent traffic delays. Delay occurs not only in signalized intersections, but also in unsignalized intersections, because delays can occur in any traffic where there is a human factor.

Some studies in the literature have focused on delays under mixed traffic conditions based on signalized intersections, and analyzes have been made about how these delays can be estimated.

In these studies, the importance of vehicle diversity in delays in traffic that include car, minibus, bus and motorized two-wheeler vehicles are investigated, the delays that may occur due to traffic signals are examined, the delays caused by signal phase timing that can cause queue length are focused, and delays are determined by using shockwave theory, analytical solutions of delays are provided and delays due to police-controlled roundabouts are assessed. (Minh et al. 2010 [7], Hillier 2015 [15], Li et al. 2016 [[16]], Mohajerpoor et al. 2019 [17] and Al-Madani 2003 [18]). As a result of these studies, the proposed method in the delay estimation of Minh et al. showed parallelism with their observed data.

In the study of Hillier et al., it was stated that delays can be minimized by optimizing traffic signals. In the study of Li et al., delays can be optimized by using the arrival rate and signal phase length parameters. In algorithms suggested by Mohajerpoor and others, signal timing formulation can be made using shockwave theory and delays can be minimized. As a result of the Al-Madani study, it is seen that the police-controlled roundabout effect is similar to that of traffic signals.

In the studies of Tian et al. (2010) [19], using traffic simulation models, the delay effects of link length, speed and vehicle headway parameters on signalized intersections were investigated and as a result of their research, shorter links and higher link speeds cause lower delays, while the effect of headway distribution on delays could not be determined exactly.

In the studies of Chandra et al. (2019) [20], the effect of heavy and light vehicle ratios in traffic on delay was examined and it was determined that heavy vehicles cause more delays. In the Ngoduy (2014) [21] study, when the delay and reaction time from the macroscopic scale were examined, it was determined how the delay of the equilibrium speed function affected the stability condition. In the study of Weng and Meng (2013) [3], delays in work zones were examined with parametric, non-parametric and simulation approaches, and it was found that the non-parametric approach was more accurate in delay estimation.

In the studies of Xi et al. (2015) [22], left-turn traffic and right-turn traffic delays were investigated using shift-share analysis methods. In another study, delay estimations were made using heuristic and machine learning algorithms.

In Lingras and Mountford's (2001) [23] study, delays were evaluated by using a genetic algorithm performing short-term inter-city traffic forecasting, and the proposed approach delay estimates in their research reflect a high degree of accuracy.

2.2. Delay Studies with Connected/Autonomous Vehicles

Connected/Autonomous Vehicles have gained popularity recently and with the orientation of many automotive companies on these vehicles, these vehicles have started to be seen more and more in traffic. Besides providing advantages in many aspects, these vehicles will not be easily adopted by society as they are different from traditional vehicles.

These vehicles have been evaluated in many ways, and the increase in the proportion of these vehicles in traffic is an indispensable fact, as it provides an advantage compared to traditional vehicles like better traffic management, because the human factor is eliminated and possible human-induced errors are minimized.

In this study, delay predictions by evaluating Connected/Autonomous Vehicles will enrich the literature, and the use of delay predictions by incident characteristics, lagged detector data, adjacent detector data & lagged adjacent detector will provide a different perspective to the literature. In this context, looking at other studies in the literature, Li et al. (2018) [24] proposed a distributed nonlinear consensus delay-dependent control algorithm to examine car following interactions, the position, velocity and acceleration/deceleration parameters in their simulations match the proposed method.

Tang and Li (2018) [25] presented the consensus-based control protocol for connected autonomous vehicles with position and velocity parameters, considering time delays in their study, and as a result of their research, it was seen that the efficiency of the proposed method is compatible with spacing error and velocity parameters.

Luan et al. (2020) [26] have proposed Model Predictive Control and Adaptive Model Predictive Control for Uncertain model algorithms by examining random network delays with the trajectory tracking model of autonomous vehicles. The algorithms they propose can ensure system stability and tracking accuracy of the autonomous vehicle under random network delays.

Gou et al. (2020) [27] have proposed the Intelligent Intersection Control Protocol using the route planning method. As a result of their studies, they showed that urban traffic can be more efficient by reducing travel times and shortening delays of autonomous vehicles. In order to solve the cooperative adaptive cruise problem in autonomous vehicles, Song et al. (2019) [28] developed a model and defined delays and demonstrated that delays could be optimized with the help of heterogeneous wireless channels.

In the study of Xu et al. (2020) [29], path tracking with delay compensation in autonomous vehicles was evaluated with 2 strategies. The experimental results show that, compared with the control that ignores hysteresis and hysteresis, the tracking error is lower and the stability margin is significantly improved.

Mehr and Eskandarian (2021) [30] evaluated delays caused by lane changing and incidents in terms of Intelligent Vehicles with traffic flow, system penetration rate, number of late-stage lane and incident duration parameters, and their proposed method has a positive impact on traffic flow and delay for intelligent vehicles. More work in literature can be found in the Table 2.1.

Table 2.1: CAV Papers in Literature.

Author(s) and Year	**Research Focus**	**Main Topic**	**Parameters**	**Method(s)**	**Region**	**Type**
Minh et al. (2010)	Delay Estimation	Delays in Traffic	2-wheeler, car, minibus and bus	Webster	Vietnam	Real Life
Hillier (2015)	Delay Minimization	Delays in Traffic	Traffic Signals	Particular Emphasis	London, England	Real Life
Li et al. (2016)	Traffic Delay	Delays in Traffic	Signal Phase Timing	Agent-Based Simulation Modelling	Detroit, USA	Real Life
Mohajerpoor et al. (2016)	Delay Minimization	Delays in Traffic	Traffic Signal Timing	Shockwave Theory	-	Literature
Al-Madani (2003)	Delay Comparison	Delays in Traffic	Roundabout and Traffic Signal	Vehicular Delay Estimation	-	Literature
Tian et al. (2010)	Delay Estimation	Delays in Traffic	Length, Free Flow Speed, Signal Timing, Spacing	Highway Capacity Manual	-	Literature
Chandra et al. (2019)	Service Delay	Delays in Traffic	Light and Heavy Vehicle Proportion	Service Delay Model	India	Real Life

Table 2.1. CAV Papers in Literature. (cont.)

Author(s) and Year	**Research Focus**	**Main Topic**	**Parameters**	**Method(s)**	**Region**	**Type**
Ngoduy (2014)	Time Delay	Delays in Traffic	FFS, Jam Density, Critical Density, Safe-time Headway, Relaxation Time, Delay Time	Treiber-type Model	-	Literature
Weng and Meng (2013)	Traffic delay	Delays in Traffic	Work Zone Capacity	Parametric non-parametrik and simulation	-	Literature
Xi et al. (2013)	Traffic Delay	Delays in Traffic	Pedestrian Volume, Left & Right Turn	Shift-Share	China	Real Life
Lingras and Mountford (2001)	Time Delay Neural Networks	Delays in Traffic	Traffic Volume	Genetic Algorithm	Alberta, Canada	Real Life

Table 2.1. CAV Papers in Literature. (cont.)

Author(s) and Year	**Research Focus**	**Main Topic**	**Parameters**	**Method(s)**	**Region**	**Type**
Li et al. (2018)	Car-Following Interactions and Heterogeneous Time Delays	Delay with Connected/ Autonomous Vehicles	Spacing Errors, Velocity, Acceleration/ Deceleration rate	Lyapunov-Krasovskii Method	-	Literature
Tang and Li (2018)	Time Delays for Non-Lane-Discipline Connected Autonomous Vehicles	Delay with Connected/ Autonomous Vehicles	Position, Velocity and Desired Gap	Lyapunov-Krasovskii method	-	Literature
Luan et al. (2020)	Random Network Delay	Delay with Connected/ Autonomous Vehicles	Road Curvature and Vehicle Speed	Model Predictive Control, Adaptive Model Predictive Control	-	Real Life

Table 2.1. CAV Papers in Literature. (cont.)

Author(s) and Year	**Research Focus**	**Main Topic**	**Parameters**	**Method(s)**	**Region**	**Type**
Gou et al. (2020)	Delay Prediction and Route Planning	Delay with Connected/ Autonomous Vehicles	Vehicle Number, Acceleration/ Deceleration, Minimum Gap and Maximum Speed	SMOTE Algorithm and iEigenAnt Algorithm	-	Literature
Song et al. (2019)	Time-delay feedback cooperative adaptive cruise control of connected vehicles	Delay with Connected/ Autonomous Vehicles	Vehicle Number, Vehicle Characteristics, Acceleration/ Deceleration and Speed	Cooperative Adaptive Cruise control (CACC)	-	Literature
Xu et al. (2020)	Time Delay	Delay with Connected/ Autonomous Vehicles	Vehicle Characteristics, Speed, Road Curvature, Yaw angle of vehicle,	Augmentation Strategy, Predictor Strategy	-	Literature

Table 2.1. CAV Papers in Literature. (cont.)

Author(s) and Year	**Research Focus**	**Main Topic**	**Parameters**	**Method(s)**	**Region**	**Type**
Mehr and Eskandarian (2021)	Traffic Delay during Freeway Incidents	Delay with Connected/ Autonomous Vehicles	Number of Lane, Incident Duration, Traffic Flow, System Penetration Rate	Sentinel	-	Literature

3. THEORY

In this section, the theory behind the machine learning methods used in this study is explained. Theoretical information about linear methods (linear regression, Bayesian ridge regression and lasso regression), regularized methods (Bayesian ridge regression, lasso regression and XGBoost), Bayesian methods (Bayesian ridge regression) and tree based non-linear methods (random forest and XGBoost) can be found under the subsections.

3.1. Linear Regression

Linear regression [31] is a popularly used regression technique due to its simplicity, interpretability and robustness. In linear regression, dependent variable (or response variable) is assumed to have a linear relationship with the given independent (or explanatory) variable or variables. Thus a change in a given independent variable results in a scaled change in the expected value of the dependent variable. The relationship between independent variables and the dependent variable can be formulated as a linear equation

$$E[y] = a_0 + a_1 x_1 + a_2 x_2 + ... + + a_M x_M = a_0 + \sum_{i=0}^{M} a_i x_i \tag{3.1}$$

or by using the matrix notation

$$Y_{1xN} = A_{1xM} X_{MxN} \tag{3.2}$$

where Y is observation vector of target variable, X is the matrix of observed dependent variables and A is the vector of model coefficients.

Since there can be an unavoidable noise in every data gathering actual instances of independent variable is modeled as

$$y = a_0 + a_1x_1 + a_2x_2 + ... + +a_Mx_M + \varepsilon = a_0 + \sum_{i=0}^{M} a_ix_i + \varepsilon \tag{3.3}$$

or in matrix form

$$Y_{1xN} = A_{1xM}X_{MxN} + E_{1xN} \tag{3.4}$$

where $\varepsilon \sim N(0, \sigma)$ i.i.d.

Meaning the observed instances of independent variable carry a normally distributed noise (error) term that is identically and independently distributed thus showing no heteroscedasticity or auto-correlation.

There are several basic assumptions of linear regression models, they can be summarized in 5 points:

(i) Linearity: The change in the value of an independent variable should be changing in a linear fashion with the change in the value of independent variable
(ii) Noise term is normally distributed with 0 mean: The asymptotic mean value of the error term should be 0 and should show a normal distribution.
(iii) Homoscedasticity: The variance of the noise term should be stable and same for all instances
(iv) No Auto-Correlation: The independent variable should not be auto-correlated.
(v) Independence of Explanatory Variables: Explanatory (Independent) variables should be linearly independent from each other, meaning their one-to-one correlations should be negligible and one independent variable should not be a linear combination of variables.

Since linear regression is used to draw a linear line that fits the given data in a decent manner, it can be thought of as finding the optimal line that goes through the given target data points in the input space. This optimal line can be thought of as the line that minimizes the total deviation (given in Figure 3.1 [32].) between the regression line and the observed data points.

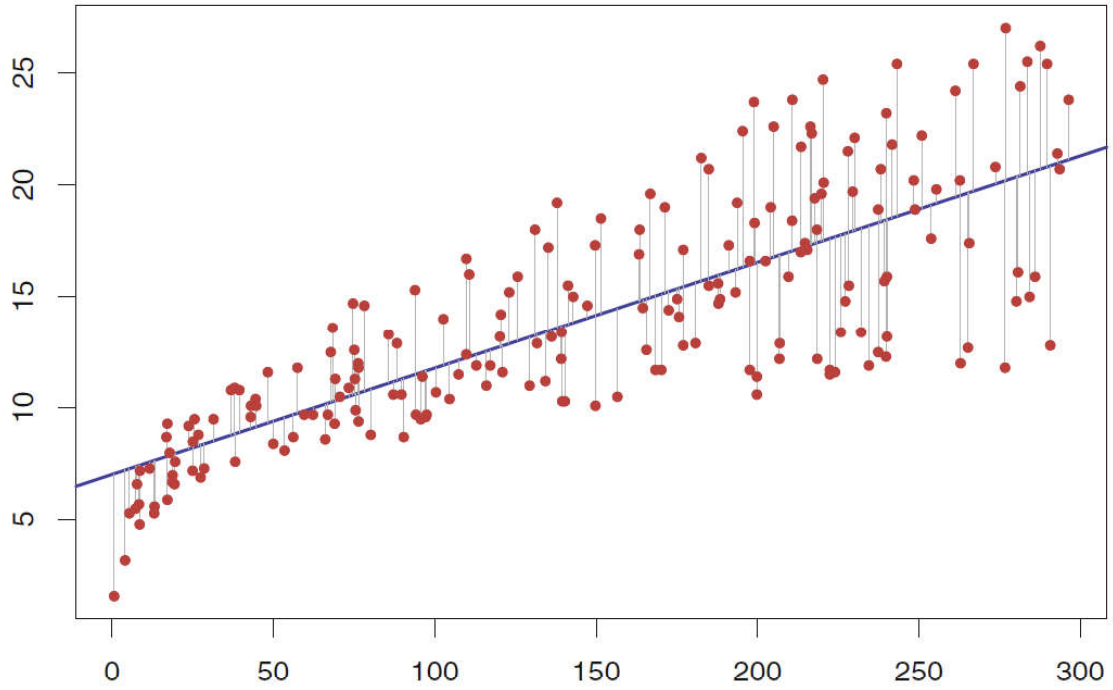

Figure 3.1. Deviations from regression line.

The cost function for the optimization problem can be taken as the sum of squared error terms where the optimized parameters are the coefficients of the linear regression. The cost function can be written as

$$\min_{X} \sum_{i=0}^{N} \varepsilon^2 = \sum_{i=0}^{N} (y_i - \sum_{j=1}^{M} a_j x_{ij})^2. \tag{3.5}$$

For the optimal value of regression parameters, the derivative of the cost function with respect to each parameter should be equal to zero as

$$\frac{\partial}{\partial a_j}\sum_{i=0}^{N}(y_i - \sum_{j=1}^{M} a_j x_{ij})^2 = 0 \ \forall a_j. \tag{3.6}$$

After taking the derivative we get

$$\frac{-2}{N}\sum_{i=0}^{N}(y_i - \sum_{j=1}^{M} a_j x_{ij}) = 0 \ \forall a_j. \tag{3.7}$$

The cost function can also be formulated in matrix form as

$$Min(Y - AX)^T(Y - AX). \tag{3.8}$$

After taking the derivative with respect to coefficient matrix A and equating the equation to 0

$$\begin{aligned} \frac{\partial}{\partial A}((Y - AX)^T(Y - AX)) &= 0 \\ 2X^TXA - 2X^TY &= 0 \\ 2X^TXA &= 2X^TY \\ X^TXA &= X^TY \end{aligned} \tag{3.9}$$

since the X^TX gives a non-singular square matrix (if the independent variable independence and no-multicollinearity assumptions are met), it can be inverted as

$$(X^TX)^{-1}(X^TX)A = (X^T)^-1X^TY \tag{3.10}$$

giving the so-called normal equation

$$A = (X^TX)^{-1}X^TY. \tag{3.11}$$

Solving the linear equation gives the optimal parameters for the regression line however, for large datasets finding the exact solution may be costly.

In such cases numerical optimization techniques such as gradient descent variants (stochastic, batch and mini-batch), adaptive moment estimation and others alike can be used to find at least locally minimal solutions.

3.2. Bayesian Ridge Regression

Bayesian Ridge Regression [33] is a Bayesian take on the ridge regression [34] which is a regularized version of multiple linear regression.

Regularization is a modelling technique that is used to overcome overfitting and also eliminate the unnecessary model parameters. Regularization techniques induce an additional regularization cost onto the model cost function for the usage of each model parameter. The greater the magnitude of the model parameter the more regularization cost it induces into the model. Thus for a model parameter to be different than 0 it should justify its value by decreasing the model cost function by achieving a greater fit more than it increases the regularization cost.

Since the regularization cost and model fitting cost can be in different magnitudes, regularization cost is multiplied by a regularization parameter before being added to the cost function as

$$Min \sum_{i=0}^{N} (y_i - \sum_{j=1}^{M} a_j x_i j)^2 + \lambda \sum_{j=0}^{M} R(a_j) \tag{3.12}$$

where R is the regularization function of choice and λ is the regularization coefficient.

The used regularization function and regularization parameter affects the found parameters. A high regularization parameter gives high emphasis on regularization cost rather than model fitting and a low regularization parameter gives too little emphasis on regularization.

L1 and L2 regularization are widely used regularization functions for linear models. L1 regularization uses the L1 norm for model parameters, meaning the absolute values of model parameters are summed as

$$L_1 = \sum_{j=0}^{M} |a_j|. \tag{3.13}$$

L2 regularization on the other hand uses the L2 norm which is equal to the squared sum of the model parameters as

$$L_2 = \sum_{j=0}^{M} a_j^2. \tag{3.14}$$

L1 regularization results in sparse results compared to L2 regularization, giving the value of 0 compared to values close to 0. Figure 3.2 shows the reason why L1 regularization gives a sparse output compared to L2 regularization in a visual sense.

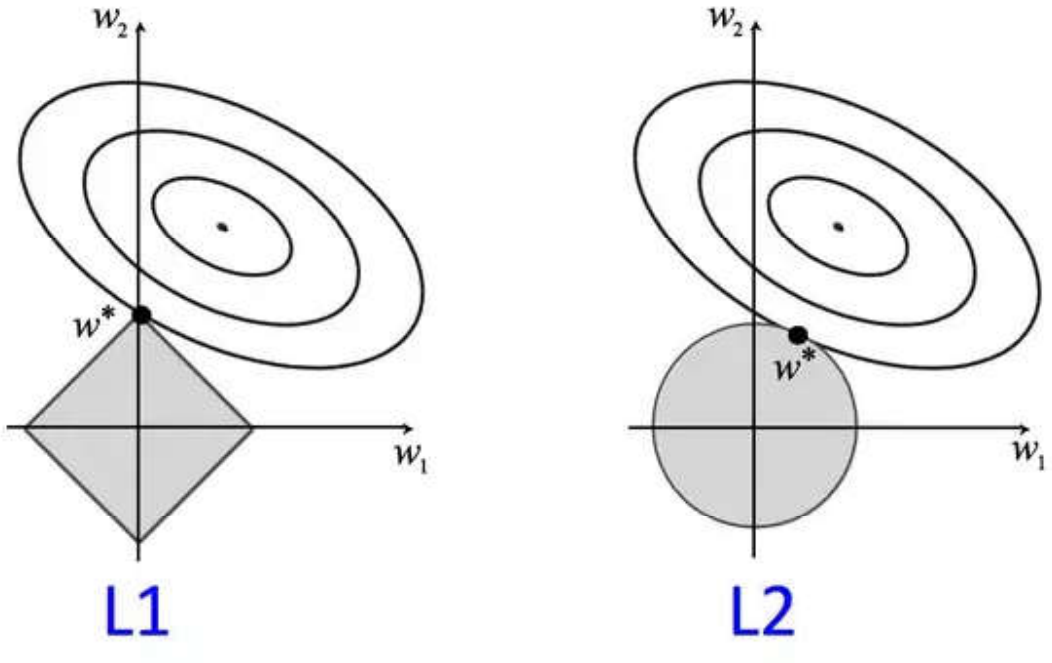

Figure 3.2. Cost function contours and regularization constraints.

If two model parameters, w_1 and w_2 are constrained to a fixed value by L1 and L2 norms, the combination of these models that minimizes the model cost (given by the contour lines) are more likely to occur on the edges of constraint space where one of the model parameters is 0.

Regularized versions of linear regression are Lasso and Ridge regressions. Lasso regression uses the L1 norm while Ridge regression uses the L2 norm. Bayesian statistics and frequentist statistics differ in the definition of probability. Bayesian statistics views probability as the degree of belief while frequentists define probability as the long term occurrence frequency.

Bayesian approach updates its prior beliefs with the upcoming data (evidence) using the Bayes theorem. Bayes theorem can be derived as

$$\begin{aligned} P(A,B) &= P(A|B)P(B) = P(B|A)P(A) \\ P(A|B)P(B) &= P(B|A)P(A) \\ P(A|B) &= \frac{P(B|A)P(A)}{P(B)} \\ P(A|B) &= \frac{P(B|A)P(A)}{\sum^{A} P(B|A)P(A)} \end{aligned} \tag{3.15}$$

where $P(A|B)$ is the conditional probability of A given B, called posterior probability, $P(B|A)$ is the conditional probability of B given A, called likelihood, $P(A)$ is the independent probability of A, called prior probability and $P(B)$ is the independent probability of B, called marginal probability.

Bayes theorem can be interpreted as if we have a prior probability for A being true, after observing B we can get a posterior probability for A, considering the likelihood of B under assumption of A is true and overall possibility of observing B.

Importance of Bayes theorem can be understood with an example. Suppose there is a rare disease, seen in the population with 0.5% probability. There is a test for that disease that gives true positives with 90% rate and true negatives with again 90% rate.

A random person tests for once and gets a positive result for the disease. Then applying the Bayes Theorem,

$$\begin{aligned} P(Sick|Positive) &= \frac{P(Positive|Sick)P(Sick)}{P(Positive)} \\ P(Sick|Positive) &= \frac{0.9 * 0.005}{0.9 * 0.005 + 0.1 * 0.995} \\ P(Sick|Positive) &= 4.33\%. \end{aligned} \tag{3.16}$$

The belief of having the disease is updated from 0.5% to 4.33%. If the same person takes the test again and still gets a positive result then

$$\begin{aligned} P(Sick|Positive) &= \frac{0.9 * 0.0433}{0.9 * 0.0433 + 0.1 * 0.567} \\ P(Sick|Positive) &\approx 40.73\%. \end{aligned} \tag{3.17}$$

The parameters in the frequentist models are assumed to be constant and unknown, they are inferred from the data using optimization techniques. Since the Bayesian approach uses beliefs rather than probabilities there is an uncertainty in model parameters. This uncertainty is induced by a probability distribution over the model parameters. A model parameter can be a certain value with a given probability thus this approach also gives a probability envelope for the predicted model results.

For the Bayesian Ridge Regression the underlying model is structured as

$$\begin{aligned} Y &= AX + \sigma^2 I \\ P(y|X, A, \sigma) &\sim N(y|AX, \sigma^2 I) \\ P(A|\tau) &\sim N(A|0, \tau^2 I) \\ \frac{1}{\sigma^2}, \frac{1}{\tau^2} &\sim G(\alpha) \end{aligned} \tag{3.18}$$

where X is the matrix of observed features, A is the coefficient matrix, σ^2 is the variance of model error, τ^2 is the variance of model parameters, N is multivariate normal distribution and G is gamma distribution.

Using the Bayes Theorem, probability of model coefficients can be written as

$$P(A|y,X) \sim P(A)P(y|A,X). \tag{3.19}$$

After expanding the probability terms it can be written as

$$P(A|y,X) \sim \exp(-\frac{1}{2}(A-0)^T\frac{1}{\tau^2}(A-0))\exp(-\frac{1}{2}(y-AX)^T\frac{1}{\sigma^2}(y-AX)). \tag{3.20}$$

After rearranging the terms we get

$$P(A|y,X) \sim \exp(-\frac{1}{2\sigma^2}(y-AX)^T(y-AX) - \frac{1}{2\tau^2}||A||_2^2). \tag{3.21}$$

Maximum a posteriori estimation for A can be written as

$$\max \exp(-\frac{1}{2\sigma^2}(y-AX)^T(y-AX) - \frac{1}{2\tau^2}||A||_2^2) \tag{3.22}$$

which can be turned into a minimization problem of the exponent term as

$$\min \frac{1}{2\sigma^2}(y-AX)^T(y-AX) - \frac{1}{2\tau^2}||A||_2^2 \tag{3.23}$$

after simplification we get

$$\min(y-AX)^T(y-AX) - \frac{\sigma^2}{\tau^2}||A||_2^2 \tag{3.24}$$

which is the same formulation with frequentist ridge regression, which uses L2 regularization given $\frac{\sigma^2}{\tau^2} = \lambda[x]$.

Using Bayesian Linear regression over multiple linear regression gives some advantages. Bayesian regression is an inherently probabilistic model and gives a probability distribution over its predictions. Its parameters are regularized, thus the model is retained from overfitting. Regularization parameter has its own probability distribution and it is inferred from the data rather than being given as a constant to the model, saving the trouble of hyperparameter tuning. Parameters are more robust to outliers due to its probabilistic nature. However, these advantages come with an increased computational cost.

Bayesian ridge regression has mostly the same assumptions as the multivariate linear regression. Other than these assumptions several assumptions are imposed on the model, which are:

(i) Model parameters come from a normal prior distribution with same variance and zero mean

(ii) Precision parameters of normal distribution (reciprocal of variance) come from a gamma prior distribution, which is the conjugate-prior for the normal distribution precision.

Starting with these uninformative prior distribution assumptions, the model tries to learn the posterior probability distribution over the parameters using the data as evidence.

3.3. Lasso Regression

Lasso regression [35] is a regularized regression technique similar to the ridge regression explained in the Bayesian Ridge Regression section. Lasso regression however, differs in its regularization method by using L1 norm instead of L2 norm. Lasso regression optimization equation can be formulated as Equation 3.25

$$Min \sum_{i=0}^{N} (y_i - \sum_{j=1}^{M} a_j x_i j)^2 + \lambda \sum_{j=0}^{M} |a_j| \tag{3.25}$$

where λ is the regularization coefficient and a_i is the coefficient calculated for feature i.

Regularization coefficient λ is the only hyperparameter of this model and should be selected using a hyperparameter tuning method. It is important to note that, unlike linear regression, Lasso regression is not scale invariant in terms of its features. Since the same regularization coefficient λ is used to penalize every coefficient and the value of coefficient is dependent on the feature scale, it is good practice to scale all features.

The advantage of Lasso regression over ridge regression is that the former gives sparse results which makes it a linear feature selection tool.

3.4. Random Forest

The building block of the random forest is a regression tree. Regression trees are simple methods that divide feature space into rectangles and assign predicted values for each rectangle. Since this method has a low bias and high variance, by ensembling multiple trees together and averaging over them, the accuracy of predictions can be increased while bias stays the same. Random Forest introduced by Breiman (2001) [36] uses this idea and has become one of the most common ML methods in predictive studies.

The idea in regression trees is dividing feature space into a number of rectangular regions and fitting simple models within these regions. We will discuss the classical case in which the simple model is a constant. Suppose we have a d-dimensional feature space denoted by X and outcome space denoted by Y. If the feature space is partitioned into M regions, there will be M parameters, predictive values for M regions, to be estimated. Formally, we can write the estimate of a point x with tree regression as

$$\hat{f}(x) = \sum_{m=1}^{M} c_m I\{x \in R_m\} \tag{3.26}$$

where c_m is the prediction for all points in region m to be estimated. Having stated this functional form, we note that the number of regions is of critical importance in this model. In the extreme scenario, if M is equal to sample size, then the tree will fit training data perfectly. However, it will perform poorly in a test set because as the number of regions increases it overfits.

It is not feasible to find the best partitioning by brute force but there is a popular greedy algorithm used in the literature. This algorithm works as follows:

(i) For each feature j, find the cutoff point s that minimizes sum of squared residuals when you choose cm as the average of outcomes in that region:

$$\sum_{x_i \in R_1} (y_i - c_i)^2 + min_{c_2} \sum_{x_i \in R_2} (y_i - c_2)^2$$

.

(ii) Find the feature that yields the minimum value from the first step

(iii) Split data from the feature and cutoff found in step (i) and (ii).

(iv) Repeat steps (i), (ii) and (iii) for each partition found in step (iii).

As this procedure continues, tree will grow until it reaches final nodes with single elements. In order to prevent overfitting, this tree has to be pruned by associating the number of end nodes with some penalty. This will stop unnecessary splitting if it doesn't increase prediction accuracy sufficiently. A common penalty is linear in the number of nodes. Say, we have a penalty parameter , then the objective function to be minimized will be

$$C_\theta(M) = \sum_{m=1}^{M} \sum_{x_i \in R_1} (y_i - \hat{c}_1)^2 + \theta |M|. \tag{3.27}$$

As θ increases, the tree will become smaller due to a larger penalty. One can choose optimal θ by cross-validation. I will discuss some remaining properties of trees in the chapter related to asymptotic results.

The idea in random forest is generating multiple trees that are uncorrelated so that when we take their average, the noise in the estimates will cancel out each other.

The random forest algorithm can be described by the following steps:

(i) Draw a bootstrap sample from the training set.

(ii) Choose k random features from d-dimensional feature space.

(iii) Grow a tree using these k features from the bootstrap sample as described in the previous section.

(iv) Repeat steps (i), (ii) and (iii), say, B times.

(v) The prediction for point $x \in X$ will be

$$\hat{f_{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(X)$$

where T_b is an individual bootstrap tree.

Since each tree is built using random features, the correlation among trees is weakened. To see this note that if a set of random variables are identically distributed with variance σ^2 and correlation ρ, then the variance of their average will become $\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$. This means that as we increase B, the second term will vanish. Also, is less than 1 because we randomly choose features in each tree. So, random forest has a lower variance in its predictions.

The critical parameter to be determined is the number of random features (k) from d-dimensional feature space. If we choose it small, then will become smaller because we will be less likely to have similar sets of features in different trees. However, using a small number of features will decrease accuracy of individual trees as well. Therefore, we have to choose k carefully by taking this trade-off into account. A rule of thumb proposes to choose $k < d$. One can also use cross-validation but it may be computationally expensive.

Until recent years, there were very few results about the asymptotic properties of random forest. Researchers were relying on the predictive performance of this method without rigorous statistical proofs. However, there have been important papers [36–39] providing results about random forest. We will not attempt to cover all the literature but just summarize some of the results.

The bias in random forest is similar to the bias in tree regression. However, we use a subspace of original feature space in each tree in the random forest.

If the true model relates outcome to all features, then each individual tree will be estimating a model with an omitted variable. This means that estimates of individual trees will be biased in such a case and aggregation over trees will not reduce this bias. If one uses subsampling instead of bootstrap and uses two different datasets to grow a tree and calculate average within a region, then the estimates become consistent and asymptotically normal around true values [37].

Friedman 2009 [39] decomposes variance of random forest estimates into two parts. One part comes from variance of individual trees and the other part comes from the correlation structure. Using this decomposition Scornet 2015 [38] proposes a way to choose an optimal number of random features from feature space when the sample size tends to infinity.

These improvements explain the statistical properties behind random forest's success in predictions but also possible problems (bias, suboptimal predictions) if we do not carefully choose the number of random features carefully.

Random forests can capture nonlinear structures in the data thanks to the nonlinear nature of trees. Furthermore, it is not susceptible to high variance in estimates because of averaging over multiple trees. A major disadvantage is that it is very difficult to interpret results of a random forest estimation. While we can observe the branching in a tree and easily understand how prediction works, one has to delve into details of estimation in order to interpret results.

3.5. XGBoost

XGBoost [40] is a special implementation of a branch of machine learning algorithms called boosting, more specifically tree boosting models. Boosting methods use an ensemble of weak learners that are built sequentially one after another. In the case of gradient boosting, the first tree tries to predict the actual target value and each new tree is built to predict the error made by the previous tree.

Each new tree's contribution to the final prediction is scaled by a learning rate in order to prevent overfitting.

The model structure can be written as

$$\hat{y}_i = f_0(X) + \sum_{i=1}^{N} \alpha f_i(X) \tag{3.28}$$

where $\hat{y}_i$ is the prediction made for instance i, f_0 is the tree fitted on target y_i, f_i sequential trees fitted on error made by previous trees and α is the learning rate.

XGBoost, on top of regular boosting methodology, adds L1 and L2 regularization on tree weights and column subsampling option similar to random forest. These regularization and randomization techniques further prevents the XGBoost model from overfitting to the given training data. Other than these model improvements XGBoost is also computationally performant thanks to its system design.

4. Data Generation Process

The data used in this study is synthetically generated using SUMO microscopic traffic simulation software. An uninterrupted 3-laned 10,500 meter long road consisting of 21 equally sized links with detectors (red dots in the Figure 4.1), which are 500 meters apart from each other, at the end of each link is built inside the model. Each lane has a width of 320 centimeters. Apart from the main road section with a length of 10,500 meters, there is an entry and an exit road section, which are both 250 meters in length. An illustration of the network can be seen below in Figure 4.1.

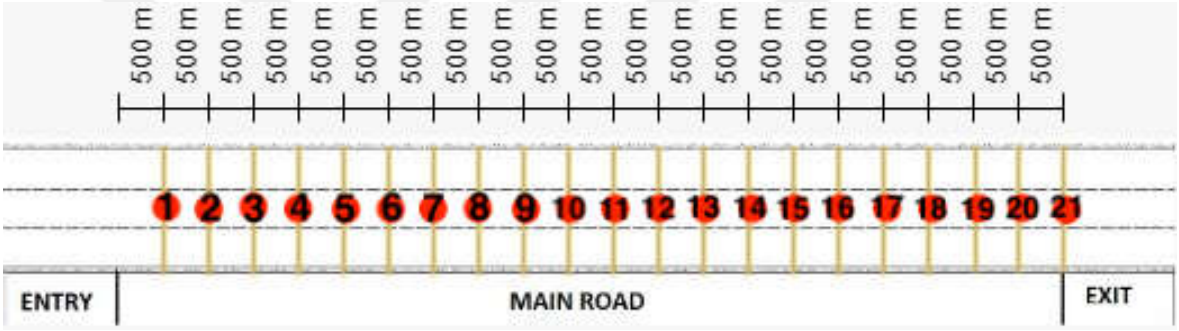

Figure 4.1. Illustration of the road network.

The given road network is used in the simulation study. Simulation parameters used in this study in SUMO microscopic simulation software are present in Table 4.1.

Table 4.1. Simulation Parameters.

Parameter	**Value**
Simulation Duration	90 minutes
Maximum Speed	110 km/hr
Detector time frequency for data collection	15 seconds

Different simulation configurations (given in Table 4.2) are runned 60 times with different seeds. An incident, if it is an incident simulation, is randomly scheduled to happen on the road between 900th and 2700th seconds of the simulation. When the time of the incident comes a vehicle in the system randomly selects a lane and a stopping point between 4500 to 8000 meters away from the starting point and stays at this position for a duration of 600 to 1500 seconds. After this stopping duration ends the said vehicle returns to its journey and the incident clears. If the simulation is not an incident simulation then none of the vehicles stop on the road on purpose.

Table 4.2. Simulation Configurations.

Configuration	**Value Range**
Incident Simulation	True, False
Incident Lane	1, 2, 3
Connected Autonomous Vehicle Ratio (%)	0, 10, 20, 30, 40, 50, 60, 70, 80
Incident Start Time (seconds)	Unif [900,2700]
Incident Duration (seconds)	Unif [600,1500]
Incident Position (meters)	Unif [4500,8000]

There are two types of vehicles present in the simulation study, which are connected autonomous vehicles and human-driven vehicles. Each of these types have different modelling parameters assigned to them. The parameters and assigned values can be seen in Table 4.3 below.

Table 4.3. Characteristics of the vehicle types.

Vehicle Parameter	**Human Driven Vehicle**	**Connected Autonomous Vehicle**
Maximum Acceleration	2.7 m/s2	2.7 m/s2
Maximum Deceleration	4.5 m/s2	4.5 m/s2
Length of vehicle	4.5 m	4.5 m
Maximum speed of vehicle	110 km/hour	110 km/hour
Speed Factor	0.9	0.9
Minimum Gap	1.5 meters	1.5 meters
Sigma	0.4	0.05
Speed Deviation	0.35	0.1

After the data is collected the first and last 15 minutes of each run is discarded as the warm-up and cool-down periods. This step is done due to the fact that simulation metrics can be misleading in this periods. After these periods are removed from the data, delay distribution is investigated for the simulation runs without any incidents.

It can be seen from the Figure 4.2 below that even when there is no accident delay times may increase or decrease depending on the timestep of the simulation. This is due to the fact that the demand given to the road is on a moderate level which creates some delay yet does not create congestion. It is also seen that there is a variance in the delay distribution which is caused by the stochasticity of the simulation.

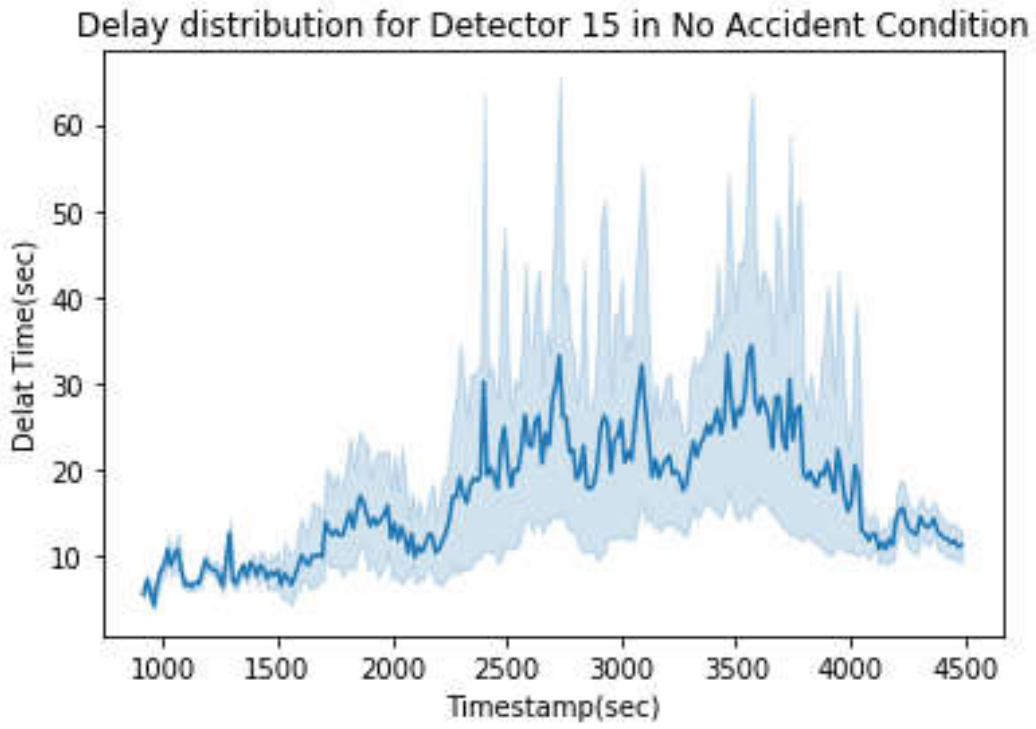

Figure 4.2. Delay Distribution for a Simulation Run with No Incident.

After investigating the simulation data for no-incident cases, simulation runs with incidents are investigated. It is seen from the data that an incident at a road section affects the upstream and downstream road sections for a couple of timesteps even after the incident is over.

It can be seen from the Figure 4.3 below that after the incident starts on road section 11 (at first blue vertical line) delay values for detector 11 starts to rise, detector 10 follows this delay increase phenomenon and after a while detector 9 shows the same behaviour. It should be also noted that the delay increase in the upstream detectors is even higher than the delay increase on detector 11. It is also observed that delay values in the downstream road section for detector 12 is lower than before the incident. After the incident clears (at second blue vertical line) delay values for all the detectors turn back to their normal levels after a substantial amount of time passes.

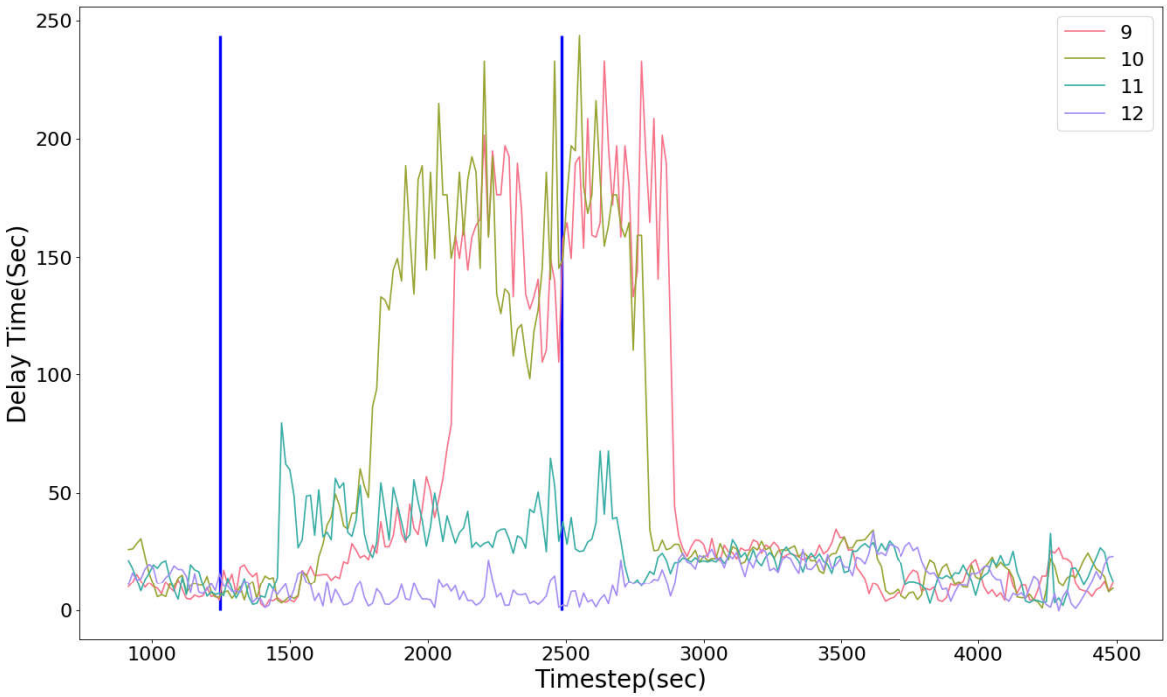

Figure 4.3. Delay vs. Timestep Plot of Detectors 9, 10, 11 & 12 for an Incident on Section 11.

After investigating these plots it can be deduced that the detector values of adjacent road sections and detector values of previous timesteps is quite informative for predicting the delay of the next timestep.

5. METHODOLOGY

In this study, delay estimation for the next timetep is modelled as a regression problem. Exploratory data analysis is conducted. New features are engineered from the insights achieved by the exploratory data analysis. Feature engineered & enriched data is split into training and testing sets using a simulation based data splitting method explained in Section 5.2. After the data is split, feature selection is applied on the training data. Different models with different hyperparameters are trained using cross validation on the selected features with the training data. The best performing model is chosen from the cross validation results. This winner model is trained on all of the training data and evaluated on the test dataset. The flowchart of the explained methodology can be found in Figure 5.1 below.

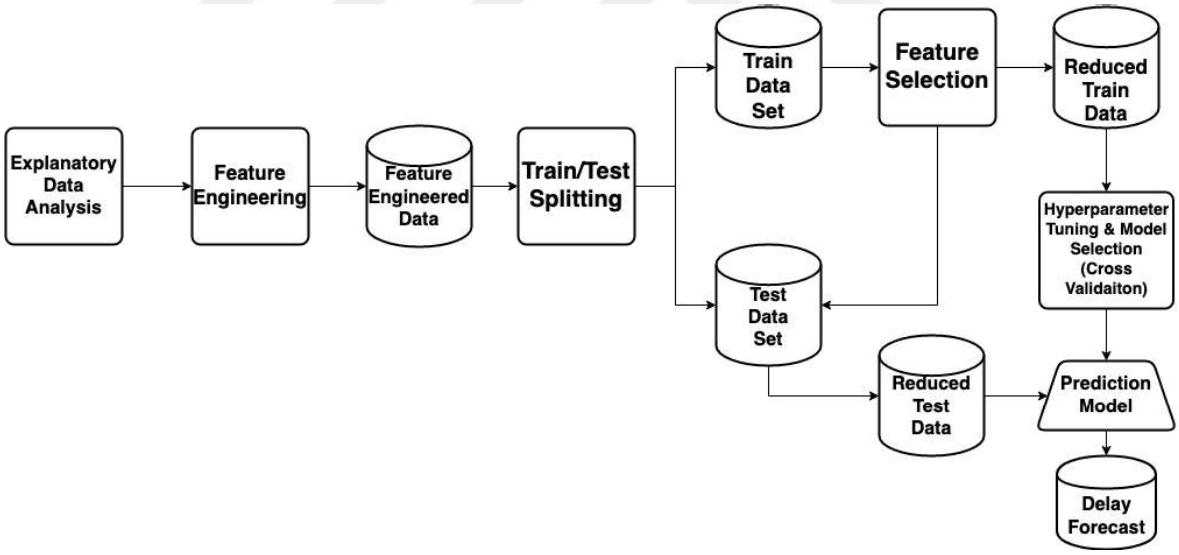

Figure 5.1. Modelling Process Flowchart.

5.1. Feature Engineering

Feature engineering can be considered as the most important part of any machine learning project since machine learning models do not understand or make judgements about the given problem but merely optimizes a given metric to the given data.

Preparing the data, transforming and crafting features relevant to the problem at hand is the duty of the modeller rather than the model. In this study, the target feature, lagged detector features and adjacent detector features are generated using feature engineering.

In time series analysis methodologies such as ARIMA or SARIMA, lagged values of the target feature are used to predict the target value in the next timestep. SARIMAX also adds exogenous variables to the model features. In this study, we use lagged values of the target similar to ARIMA, exogenous variables similar to SARIMAX and also add lagged values of exogenous variables.

5.1.1. Target

The main purpose of this study is predicting the future (next timestep) delay based on the data up to current time period. Thus, section travel time delay is needed. Travel time delay is defined as the excess amount of time spent traveling on a road section compared to a vehicle passing through the same road section with the free flow speed. Travel time delay is calculated with equation

$$D_{i,t} = \frac{L_i}{V_f} - T_{i,t} = FT_i - T_{i,t} \tag{5.1}$$

where $D_{i,t}$ is the delay on road section i in time period t, L_i is the length of road section i, $T_{i,t}$ is the actual travel time on road section i in time period t and FT_i is the free flow travel time on road section i.

Free flow travel time is estimated from the simulation data using the minimum travel time recorded in simulation runs without any road incidents. Determined free flow travel time is subtracted from actualized average travel time recorded by the detectors to find the actualized delay in each timestep.

For each simulation run, each detector and each timestep, calculated delay of the next timestep is used to create the target column.

After this process, the last timestep for each detector has an empty target column. These data points with empty target columns are dropped from the training data.

5.1.2. Lagged Detector Features

Lagged features are created for capturing the time varying dynamic effects similar to time series analysis. Flow, density, average speed, section travel time and section travel time delay are lagged by 1 up to 10 timesteps and a new column is created in the data for each detector metric lag combination. The generated lagged features can be found in Table 5.1 below.

Table 5.1. Original & Lagged Features.

Original Feature	**Lagged Features**
flow_vehph	flow_vehph_lag1, flow_vehph_lag2, ... , flow_vehph_lag10
density_vehpkm	density_vehpkm_lag1, density_vehpkm_lag2, ... , density_vehpkm_lag10
avg_speed_kmph	avg_speed_kmph_lag1, ... , avg_speed_kmph_lag10
section_travel_time_sec	section_travel_time_sec_lag1, ... , section_travel_time_sec_lag10
delay_time_sec	delay_time_sec_lag1, delay_time_sec_lag2, ... , delay_time_sec_lag10

After creating these lagged features, empty values occurred in the initial timesteps. These empty lag feature values are backfilled by the first non-empty value by assuming uniformity in the simulation time boundaries.

5.1.3. Adjacent Detector Features

Adjacent features for the next and previous detectors are created for capturing the upstream and downstream effects on the observed detector. Flow, density, average speed, section travel time, section travel time delay and lagged values of all these features for the next and previous detectors are created.

After creating these adjacent detector features previous detector features for the first detector and next detector features for the last detector had empty values.

Similar to the lagged features, uniformity assumption is applied and these empty values are backfilled and forward-filled for the first and last detector, respectively.

It is important to note that since adjacent detector features are created on top of the lagged detector features, lagged values of these adjacent features are also included in the data. The generated adjacent detector features can be found in Table 5.2 below.

Table 5.2. Original & Adjacent Features.

Original Feature	**Next Detector Features**	**Previous Detector Features**
flow_vehph	next_detector_flow_vehph	previous_detector_flow_vehph
density_vehpkm	next_detector_density_vehpkm	previous_detector_density_vehpkm
avg_speed_kmph	next_detector_avg_speed_kmph	previous_detector_avg_speed_kmph
section_travel_time_sec	next_detector_section_travel_time_sec	previous_detector_section_travel_time_sec
delay_time_sec	next_detector_delay_time_sec	previous_detector_delay_time_sec
flow_vehph_lag1	next_detector_flow_vehph_lag1	previous_detector_flow_vehph_lag1
...	...	...
delay_time_sec_lag10	next_detector_delay_time_sec_lag10	previous_detector_delay_time_sec_lag10

5.1.4. Incident Related Features

Incident related features are added to the feature space to be aware of the location and situation of the incident. Incident related features can be listed as

- Is there an accident in this timestep?
- Time passed since the accident occured (0 if there is no accident)
- Distance to the accident point (Infinite if there is no accident)
- Accident lane (Lane 1, 2, 3 for right, middle & left lanes respectively)
- Connected Autonomous Vehicle Ratio

5.2. Simulation Based Data Splitting

Simulation data is used for training the model, tuning model hyperparameters and assessing the trained and tuned model's performance. Data needs to be split into 3 mutually exclusive sets named training, validation and test sets for training, tuning and testing.

Mutual exclusivity is important since carrying over a knowledge to the training set that will not be available in the real world application creates an unrealistic and misleading performance. In order to prevent any data leakage, data needs to be split carefully before starting the modelling process.

If each row in a given tabular data is independently and identically distributed, randomly selecting the rows of data into training, validation and test sets would be possible. However, in our case each data point is correlated to neighbouring timesteps and data characteristics depend on simulation configurations.

Due to this condition, data is first split into groups based on their discrete simulation configurations, which are accident lane, existence of accident and connected autonomous vehicle ratio in traffic mix. Splitting the data based on configurations makes sure that both training and testing sets are representative of the actual data distribution.

After the data is split into groups, simulation runs in each group are selected into train and test sets using a 80% and 20% ratio respectively. Training and testing data sets are constructed by aggregating the data in selected simulation runs. This simulation run based data splitting decorrelates the training and testing sets from each other since each simulation run is independent from each other. After the data is split into training and testing data sets, the training set is further divided into training and validation sets during the cross validation step using the same methodology. The explained steps of the followed data splitting methodology can be viewed in the Figure 5.2.

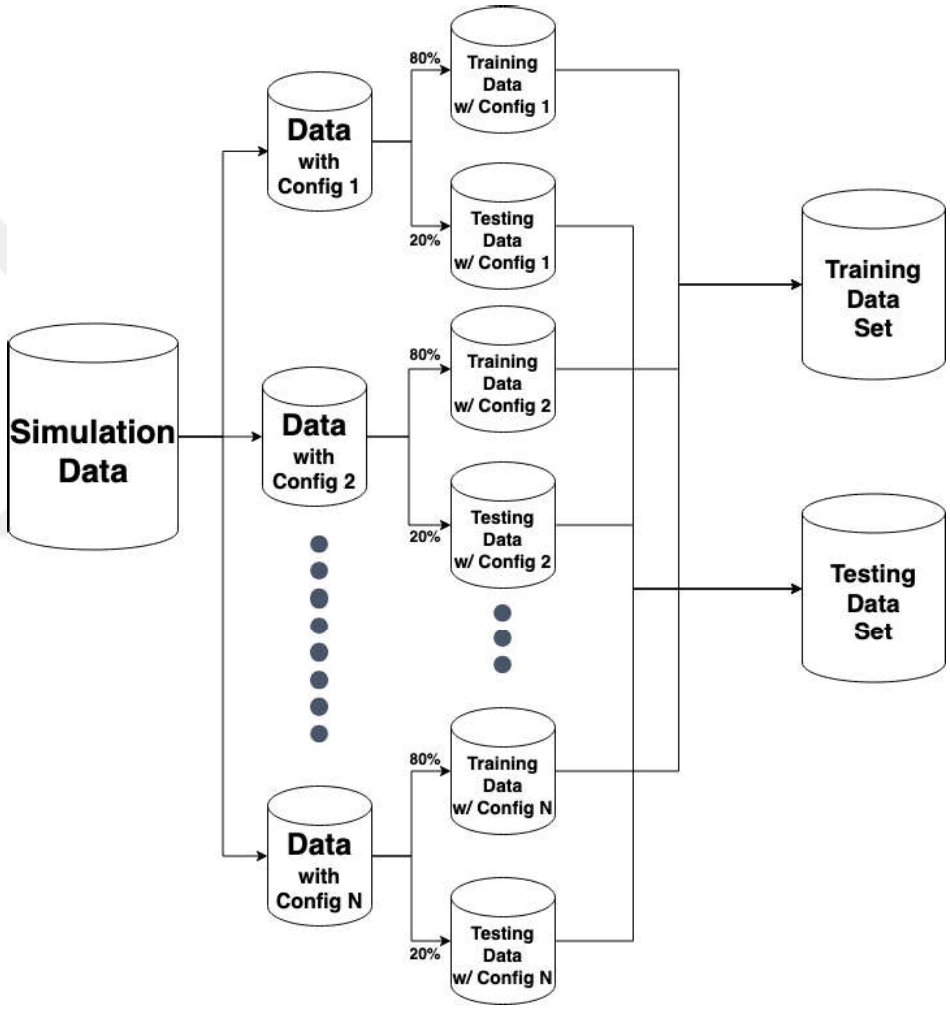

Figure 5.2. Simulation Based Data Splitting Process.

5.3. Feature Selection Process

Feature selection is an important step in machine learning model building. Relevant features are selected and others are discarded. This step decreases the dimensionality of features, lowers the computation times and improves the model convergence. 3 different feature selection metrics are used in this study, which are correlation analysis, mutual information criterion and regularized & standard random forest feature importance.

5.3.1. Correlation Analysis

Correlation analysis between two random variables X and Y gives the correspondence of Y to a change in variable X. The degree of this correspondence is measured with the correlation coefficient calculated with the Pearson correlation equation as

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}}. \tag{5.2}$$

Correlation coefficient not only quantifies the degree of relationship but also gives the nature of the relationship. Positive correlation coefficient indicates that as X increases, Y also tends to increase and the straight opposite is true for the negative correlation coefficient.

One downside of correlation analysis is that, as it can be deduced from the equation, correlation coefficient only measures linear relationships and underestimates or completely misses the nonlinear relationships. Correlation coefficient for various linear and nonlinear relationships can be found in Figure 5.3 [41] below. It can be seen that variables in horizontal and vertical axes depend on each other with a nonlinear equation; however, correlation coefficient turns out to be 0.

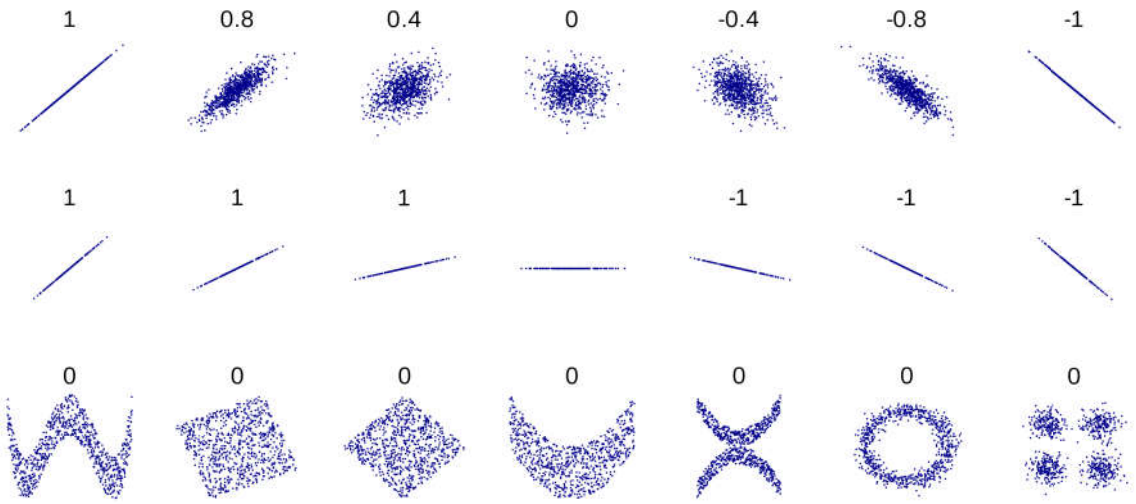

Figure 5.3. Correlation Coefficient For Different Relationships.

5.3.2. Mutual Information Criterion

The concept of information entropy plays an important role in mutual information. Entropy of a random variable quantifies the inherent degree of uncertainty in the value of the variable's outcome. High entropy means the tails of the variable's distribution is spreaded out.

Uniform distribution for example has a higher entropy compared to a normal distribution with the same mean value since it is more likely to see outcomes closer to the mean value for the normal distribution yet every value is equally likely for the uniform distribution. The information entropy is calculated as

$$H(X) = -\sum_{i} p(x_i) \log p(x_i). \tag{5.3}$$

Conditional entropy of Y conditioned on X gives the degree of uncertainty in the value of Y given the value of X. Conditional entropy of two random variables is calculated similarly as

$$H(Y|X) = -\sum_{i}\sum_{j} p(y_i, x_j) \log \frac{p(y_i, x_j)}{p(x_j)} \tag{5.4}$$

mutual information [42] between two random variables X and Y gives the reduction in information entropy of Y after observing the value of X. Mutual information is calculated by

$$I(X;Y) = H(Y) - H(Y|X) = \iint p(x,y) log(\frac{p(x,y)}{p(x)(y)}) dxdy. \tag{5.5}$$

Mutual information can capture one-to-one nonlinear relationships as well as the linear ones. However, it does not give the nature of relevance between the variables like the correlation coefficient does.

Downside of mutual information is that since it is calculated between two variables X and Y, mutual information may not capture cases where X cannot solely reduce the uncertainty of Y, yet knowing two variables X1 and X2 can reduce the uncertainty of Y. One such example is the "exclusive or" (XOR) problem shown in Figure 5.4 [43] below which can be stated as

$$Y = XOR(X_1, X_2) \tag{5.6}$$

where $XOR(1,1) = 0$, $XOR(0,0) = 0$, $XOR(1,0) = 1$ and $XOR(0,1) = 1$.

In XOR problem, knowing X1 or X2 alone does not reduce the uncertainty in the value of Y, however, knowing both of them reduces the uncertainty to 0.

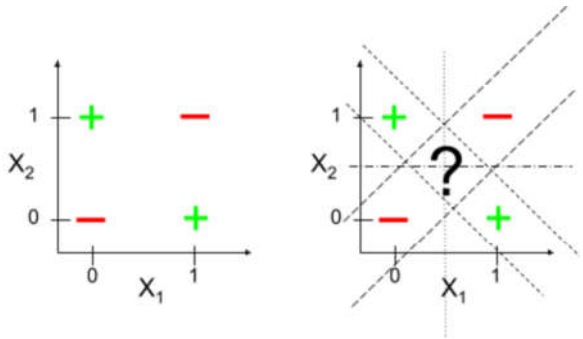

Figure 5.4. XOR Problem.

5.3.3. Regularized & Standard Random Forest Feature Importance

Random forest is an ensemble machine learning algorithm that trains many decision or regression trees (hence the name forest) on randomly sampled (bagged) data sets with randomly sampled features. This data and feature randomization results in a more robust model compared to regular tree models.

Trees inside the random forest use an impurity measure to decide on which split to make. User impurity measure may be Gini index or information gain (reduction in entropy) for classification and decrease in mean standard error or mean absolute error for regression. Whatever impurity measure is used, the features that reduce the impurity the most when a split is made are the most important features for the model. Feature importance values are calculated for random forest by taking the average decrease in impurity generated by a split in a given feature in each tree, which contains that feature in its feature set.

Random forest is also robust to the XOR problem mentioned on Mutual Information Criterion since it randomly samples the data, considers all of the features at once and splits sequentially.

One downside of random forests and decision trees is that they make their splits greedily, which sometimes results in redundant features in the feature set. Using regularized random forest [44] over standard random forest, like other regularization methods, gives sparse results by applying a regularization method on the split decision. Regularized random forest discounts the entropy reduction with a regularization coefficient $\lambda \in [0, 1]$ if the given split introduces a new feature to the feature set. If the feature used by the split is already in the feature set no discounting is applied. This discounting discourages the algorithm from adding new features to the feature set if their entropy reduction is not high enough.

5.3.4. Feature Selection

Features are ranked in descending order by their relevance (importance) scores calculated by mutual information (MI), correlation analysis & regularized random forest feature importance (RRF FI) & standard random forest feature importance (RF FI).

After the ranking, relevance metrics are merged together, any feature that is ranked under the 30th place in any of these metrics is selected for the training data. It can be observed that detector data coming from the next detector and the current detector are found to be more important through more lag periods compared to the data coming from the previous detector. It is an expected result since the delay is more influenced by downstream events compared to the upstream events. All of the selected features can be found in Table 5.3.

Table 5.3: Relevance Ranks for Selected Features.

Feature Name	**Correlation Rank**	**MI Rank**	**RF Feature Importance Rank**	**RRF Feature Importance Rank**
section travel time sec	2	1	1	2
delay time sec	1	2	2	1
avg speed kmph	46	3	3	3
avg speed kmph lag1	53	4	6	6
section travel time sec lag1	4	5	5	4
delay time sec lag1	3	6	4	5
prev detector avg speed kmph lag3	90	7	71	74
prev detector section travel time sec lag3	93	8	62	65

Table 5.3. Relevance Ranks for Selected Features. (cont.)

Feature Name	**Correlation Rank**	**MI Rank**	**RF Feature Importance Rank**	**RRF Feature Importance Rank**
prev detector delay time sec lag3	94	9	63	69
avg speed kmph lag2	56	10	18	20
prev detector avg speed kmph lag4	95	11	107	99
density vehpkm	21	12	10	11
prev detector delay time sec lag2	78	13	72	50
prev detector section travel time sec lag2	77	14	35	47
section travel time sec lag2	12	15	11	12
delay time sec lag2	11	16	12	9
prev detector avg speed kmph lag2	87	17	41	46
prev detector delay time sec lag4	99	18	110	114
prev detector section travel time sec lag4	98	19	124	149
section travel time sec lag3	5	20	27	17
delay time sec lag3	6	21	17	18
avg speed kmph lag3	58	22	25	25
prev detector avg speed kmph lag1	88	23	111	117

Table 5.3. Relevance Ranks for Selected Features. (cont.)

Feature Name	**Correlation Rank**	**MI Rank**	**RF Feature Importance Rank**	**RRF Feature Importance Rank**
prev detector delay time sec lag1	68	24	88	145
prev detector section travel time sec lag1	67	25	109	147
avg speed kmph lag4	60	26	47	100
density vehpkm lag1	26	27	22	16
prev detector avg speed kmph lag5	106	28	82	120
prev detector density vehpkm lag3	89	29	74	84
delay time sec lag4	8	33	58	55
section travel time sec lag4	7	36	24	54
delay time sec lag5	9	38	80	73
section travel time sec lag5	10	39	83	82
delay time sec lag6	15	44	96	96
density vehpkm lag2	27	45	16	19
section travel time sec lag6	16	46	119	121
section travel time sec lag7	18	51	76	91
delay time sec lag7	17	52	52	63
section travel time sec lag8	25	57	28	22
delay time sec lag8	24	58	20	23

Table 5.3. Relevance Ranks for Selected Features. (cont.)

Feature Name	**Correlation Rank**	**MI Rank**	**RF Feature Importance Rank**	**RRF Feature Importance Rank**
avg speed kmph lag8	70	59	21	24
next detector avg speed kmph lag1	108	69	32	29
next detector section travel time sec lag1	50	75	29	31
next detector avg speed kmph lag5	101	85	13	14
next detector section travel time sec lag4	28	93	48	51
next detector delay time sec lag4	29	97	40	41
next detector section travel time sec lag6	14	99	8	8
next detector delay time sec lag6	13	100	7	7
next detector section travel time sec lag5	19	105	14	13
next detector delay time sec lag5	20	106	15	15
next detector avg speed kmph lag6	100	107	9	10
next detector section travel time sec lag7	23	112	117	101
next detector delay time sec lag7	22	113	121	127

Table 5.3. Relevance Ranks for Selected Features. (cont.)

Feature Name	**Correlation Rank**	**MI Rank**	**RF Feature Importance Rank**	**RRF Feature Importance Rank**
next detector density vehpkm lag5	65	117	23	26
next detector density vehpkm lag6	69	119	30	28
next detector density vehpkm lag8	74	124	39	27
next detector flow vehph lag4	160	157	19	21
next detector flow vehph lag8	163	159	26	32

5.4. Considered Models

Different machine learning models have different advantages & disadvantages and determining the required model beforehand is not possible most of the time. Various machine learning models are trained and applied on the training and validation sets for the regression task at hand and their performances are considered.

Linear models such as linear regression, bayesian ridge regression and lasso regression are chosen for their simplicity and explainability. Random Forest is chosen for its versatility, parallelizability, ability to capture nonlinear and multivariate relationships, robustness to overfitting and feature interpretability. XGBoost is chosen for its historical success in various machine learning problems.

5.5. Cross Validation, Hyperparameter Tuning & Model Selection

5-repeated 5-fold cross validation methodology is used in the training and tuning of the mentioned machine learning models as well as in evaluating their performances. Subsection 5.5.1 explains the optimized hyperparameters of each model and gives their search spaces. Subsection 5.5.2 gives the details of the used cross validation methods and how its results are interpreted in order to choose the best performing model.

5.5.1. Hyperparameter Space of the Models

Important hyperparameters for each model are determined and a considerably large search space is defined for each of these parameters. Optimized hyperparameters and their search space can be observed in Table 5.4. Generally, as the model complexity grows the number of hyperparameters to be determined increases. Linear regression does not have any hyperparameters. For the Lasso regression, the single most important hyperparameter is regularization coefficient (alpha in Table 5.4).

Bayesian Ridge regression on the other hand has 4 important hyperparameters which are shape and scale parameters for prior distribution of model's error's standard deviation (alpha_1 and alpha_2 in Table 5.4, respectively) and shape and scale parameters for prior distribution of model coefficients' standard deviation (gamma_1 and gamma_2 in Table 5.4, respectively).

Random forest has many hyperparameters to select and tune; however, most of these hyperparameters share some common objectives and setting several of them give satisfying results. Number of trees to be trained (n_estimator in Table 5.4) determines the number of trees to be trained on the given input data. Generally setting the number of estimators too high does not decrease the accuracy of the model however, may lead to longer training times. Setting the number of estimators too low may result in a model with higher variance and higher bias thus a modeller needs to set the number of trees to be trained sufficiently high.

Minimum number of samples required to be a leaf node (min_samples_leaf in Table 5.4) adjusts the depth of trees trained in random forest. This depth adjustment is done in order to prevent overfitting and also underfitting. Number of features considered during making a split (max_features in Table 5.4) determines how many of the features are randomly selected for each tree in the random forest to train on. This hyperparameter adjusts the randomization and diversification of the trees inside the random forest.

XGBoost, similar to the Random Forest, has many hyperparameters to select and tune and again similar to the Random Forest most of these hyperparameters serve similar purposes and tuning a couple of them gives good results. Number of trees to be trained (n_estimator in Table 5.4) gives the number of trees to be trained sequentially. Due to the sequential nature of boosting algorithms, unlike random forest, setting this parameter high may increase the overfitting and may result in models with higher variance.

Learning rate (learing_rate in Table 5.4) is the discounting factor for each additional tree's input, like in many optimization algorithms setting the learning rate too high or too low hurts the performance. Minimum child weight is similar to the "minimum number of samples required to be a leaf node" parameter in random forest. This hyperparameter adjusts the depth of the built trees' branches. Finally alpha (reg_alpha in Table 5.4) is the regularization parameter on the leaf nodes of the regression trees.

5.5.2. Cross Validation & Model Selection

5-Repeated 5-Fold cross validation with randomized parameter search is used to tune the model hyperparameters inside their search space and assess the tuned model performances. In each repetition, the training data set is randomly split into 5 folds using the splitting method explained in Section 5.2. For each fold, the other 4 data partitions are used to train the model and the remaining fold is used for measuring the test performance. After the model is trained and evaluated for each fold another repetition begins with another random data split.

Table 5.4. Model Hyperparameters and Search Spaces.

Model	**Hyperparameter**	**Explanation**	**Search Space**
Linear Regression	-	-	-
Lasso Regression	alpha	Regularization Coefficient	[0,10]
Bayesian Ridge Regression	alpha_1	Shape parameter of prior inverse gamma distribution for model error's standard deviation.	[0,1]
	alpha_2	Scale parameter of prior inverse gamma distribution for model error's standard deviation.	[0,1]
	gamma_1	Shape parameter of prior inverse gamma distribution for model coefficients standard deviation.	[0,1]
	gamma_2	Scale parameter of prior inverse gamma distribution for model's coefficients' standard deviation.	[0,1]
Random Forest Regressor	n_estimators	Number of trees to be trained	100, 500, 1000, 1500
	min_samples_leaf	Minimum number of samples required to be a leaf node	1, 10, 50, 100, 200, 500
	max_features	Number of features considered during making a split	"auto", "sqrt", "log2", 1/3
XGBoost Regressor	n_estimators	Number of trees to be trained	100, 500, 1000, 1500
	learning_rate	Learning rate for each additional tree's output in sequence	0.01, 0.03, 0.05, 0.1, 0.2, 0.3
	min_child_weight	Minimum sum of instance weight required to be a leaf node	1, 10, 50, 100, 200, 500
	reg_alpha	L1 regularization term on leaf node weights	0, 0.05, 0.1

The splitting algorithm can be summarized in pseudocode given in Figure 5.5.

```
Set number of repetitions N ← 5
Set number of folds K ← 5
Get the candidate models with their hyperparameters
for n = 1 to N do
  Randomly Split data into K folds
  for m, p in model/hyperparameter combinations do
    for k = 1 to K do
      Train model m with hyperparameter set p on remaining K − 1 folds
      Calculate the performance metric of trained model on current fold k
      Record the training and testing performance on fold k
    end for
    Record the average training and testing performance for model m with hyperparameters p
  end for
  Record the cross validation performances for repetition n
end for
Return the recorded cross validation results
```

Figure 5.5. 5 Repeated 5-Fold Cross Validation Pseudocode.

Since delay estimation is a regression problem, the sum of squared errors (SSE) is chosen as the performance metric. The SSE is calculated using the Equation 5.7.

$$SSE = \sum_{i}^{N} \varepsilon_i^2 = \sum_{i}^{N} (y_i - f(X))^2 = \sum_{i}^{N} (y_i - \hat{y}_i)^2 \tag{5.7}$$

where ε_i^2 is the square of error made for instance i, y_i is the true target value for instance i, X is the set of features used by the model, $f(X)$ is the model function that maps the given input features to estimation and $\hat{y}_i$ is the predicted target value for instance i.

Mean sum of squared errors for each model in each repetition is given in Table 5.5. It can be observed that on average best average model performance is achieved by XGBoost with lower standard deviation between repetitions compared to other models. However, random forest in repetition 3 outperforms all other models. The close competition between XGBoost and Random Forest is an expected result since it is shown in the literature [45, 46] that boosted trees and random forest ensemble models give comparable results if appropriate hyperparameters are chosen.

Table 5.5. Cross Validation Results.

Model	**Repetition 1**	**Repetition 2**	**Repetition 3**	**Repetition 4**	**Repetition 5**	**Mean**	**STD**	**Best**
Linear Regression	486.95	465.38	535.35	468.02	473.63	485.87	28.89	465.38
Lasso Regression	2024.87	1766.99	1901.95	1640.76	1759.25	1818.76	147.74	1640.76
Bayesian Ridge Regression	486.94	465.38	535.35	468.01	473.64	485.86	28.89	465.38
Random Forest Regressor	373.4	339.69	373.24	390.27	403.48	376.02	23.94	339.69
XGBoost Regressor	368.95	360.27	400.13	364.94	354.19	369.70	17.88	354.19

The random forest is chosen as the final model since it gave the lowest sum of squared error among other model & hyperparameter combinations for hyperparameters listed in Table 5.6.

Table 5.6. Hyperparameters for Random Forest Regressor.

Hyperparameter	**Value**
n_estimators	100
min_samples_leaf	10
max_features	1/3

6. Results

In this section the prediction results obtained by the chosen machine learning model is shown. Validity of the finalized model's predictions is investigated using the permutation feature importance and SHAP values. At last the prediction performance of the model and the effects of autonomous connected vehicles in varying simulation conditions and traffic mixes is discussed.

6.1. Model Results

After the best performing model with best fitting hyperparameters is determined, this model is once again trained on the entire training dataset. After training, the model's performance is evaluated on training and testing data sets. Final model had a mean squared error value of 224.89 for training and 247.77 for testing datasets. The slight increase in mean squared error for the testing data set is expected and since there is not a major distance it can be said that the model neither overfits nor underfits to the given data.

After the model training, permutation feature importance with 10 repetitions is calculated for the trained model. The mean decrease in mean squared error for each used feature can be observed in Figure 6.1 below. It can be observed that the model determined section_travel_time_sec, delay_time_sec and avg_speed_kmph as the three most informative features for the regression problem. These three features were also selected as the top three most relevant features by the mutual information criterion and regularized & standard random forest feature importance. Lagged features of main, previous and next detectors are also found to be important compared to the others. Connected autonomous vehicle percentage in the traffic flow is also found to be the most important outside condition. The exact permutation feature importance values can be found in Appendix A.1.

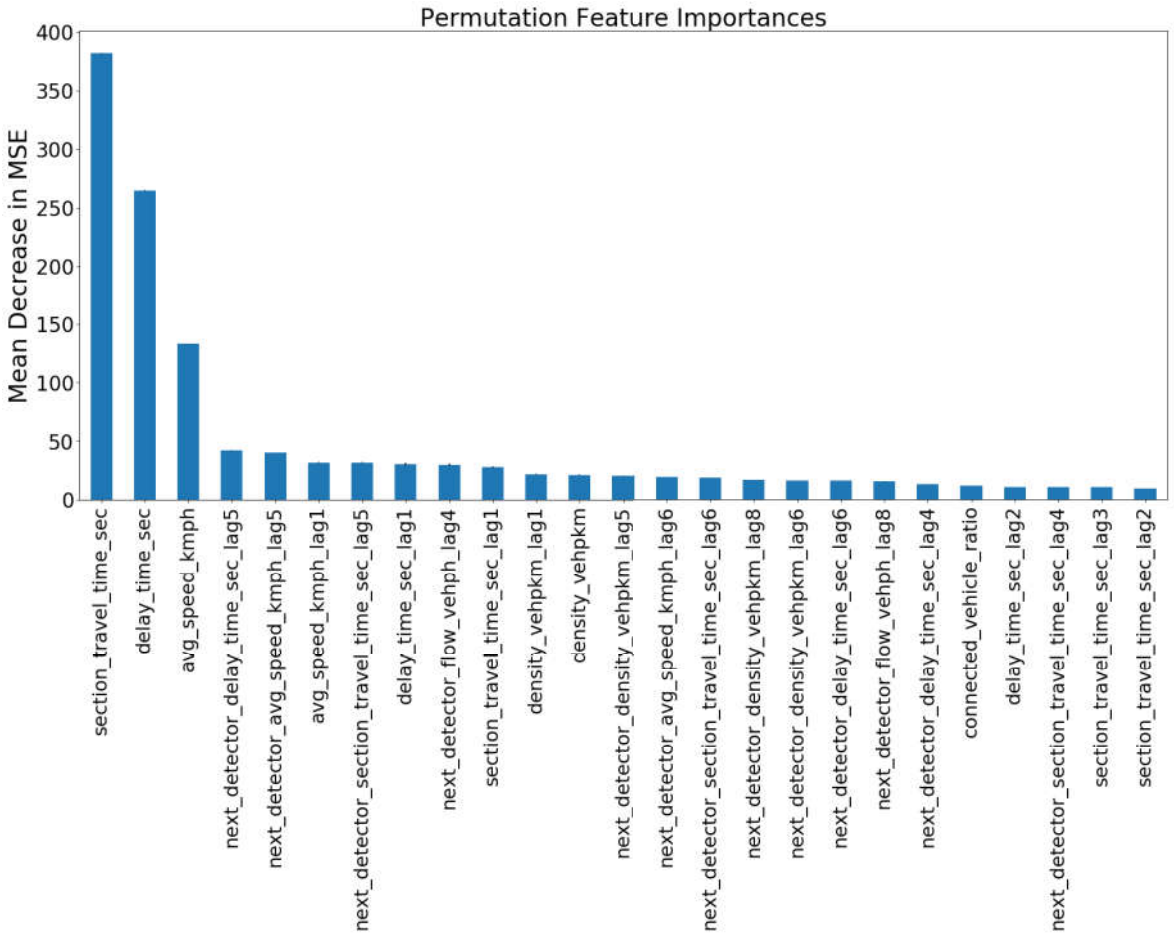

Figure 6.1. Permutation Feature Importance for Selected Model.

After the inspecting the permutation feature importance values, the SHAP values [47] of the model features is investigated. The SHAP value of a feature gives the individual contribution of this feature in the model's predictions. SHAP values do not give causality or do not necessarily show correlations in the data but shows the effect of the given feature in the model's behavior.

It is observed from the Figure 6.2 that high values of section_travel_time has a positive and big contribution in the delay prediction. It is seen that the SHAP values of CAV ratio is dwarfed compared to other features.

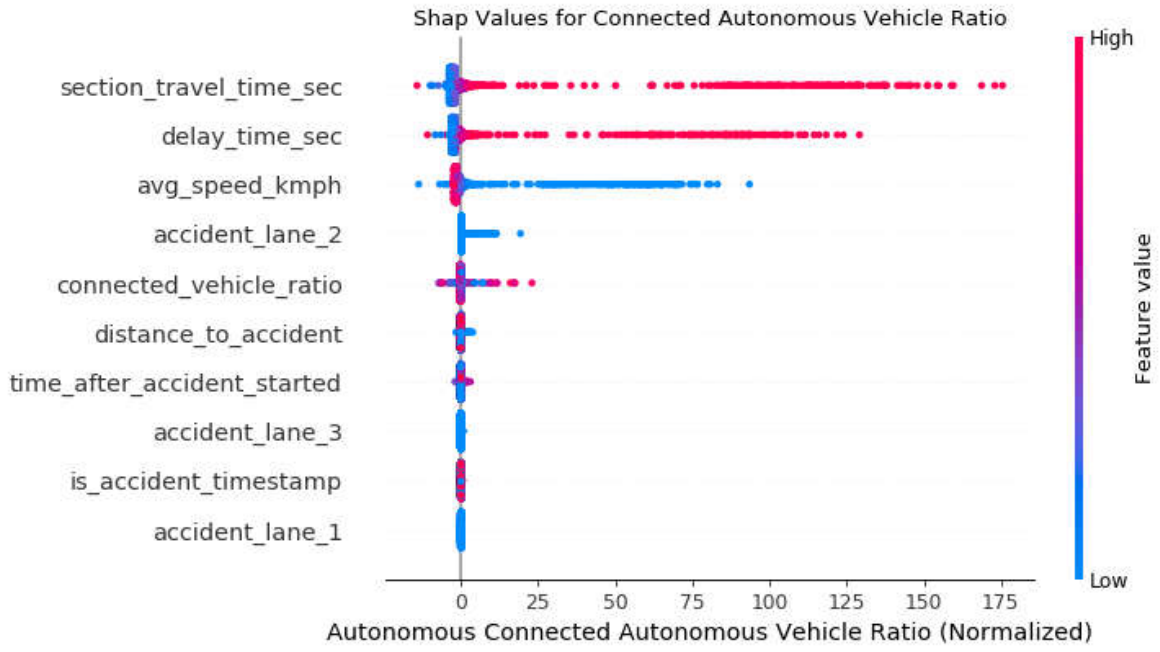

Figure 6.2. Shap Values Random Forest Model.

The SHAP values of CAV ratio is inspected on a closer look in Figure 6.3 by changing the scale of the x axis. It is seen that, for most of the predictions, the SHAP values of CAV ratio is close to 0 and there is a variance. The SHAP values for different levels of CAV ratio can be found in Figure 6.4. It can be deduced that as the CAV ratio increases the median SHAP value decreases slightly, meaning the increased CAV ratio, on avarage, results in slightly decreased delay predictions.

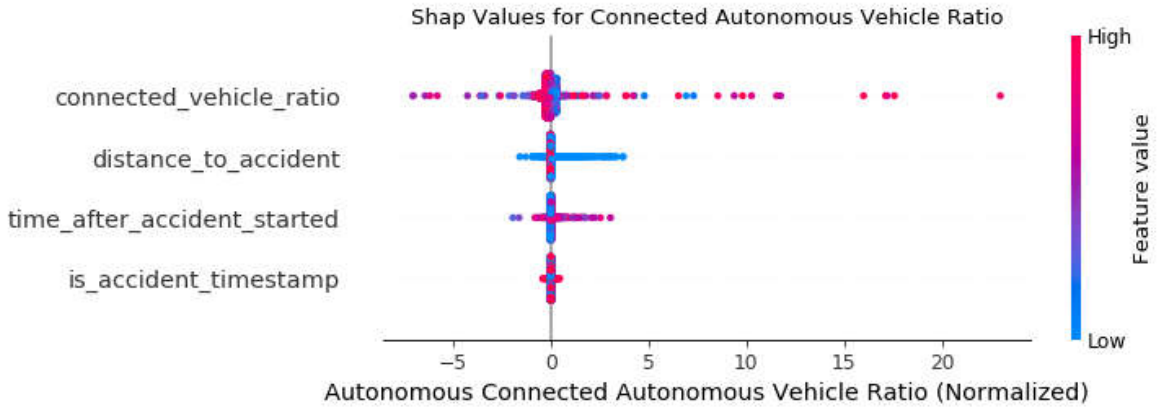

Figure 6.3. Shap Values for CAV Ratio Feature of Random Forest Model.

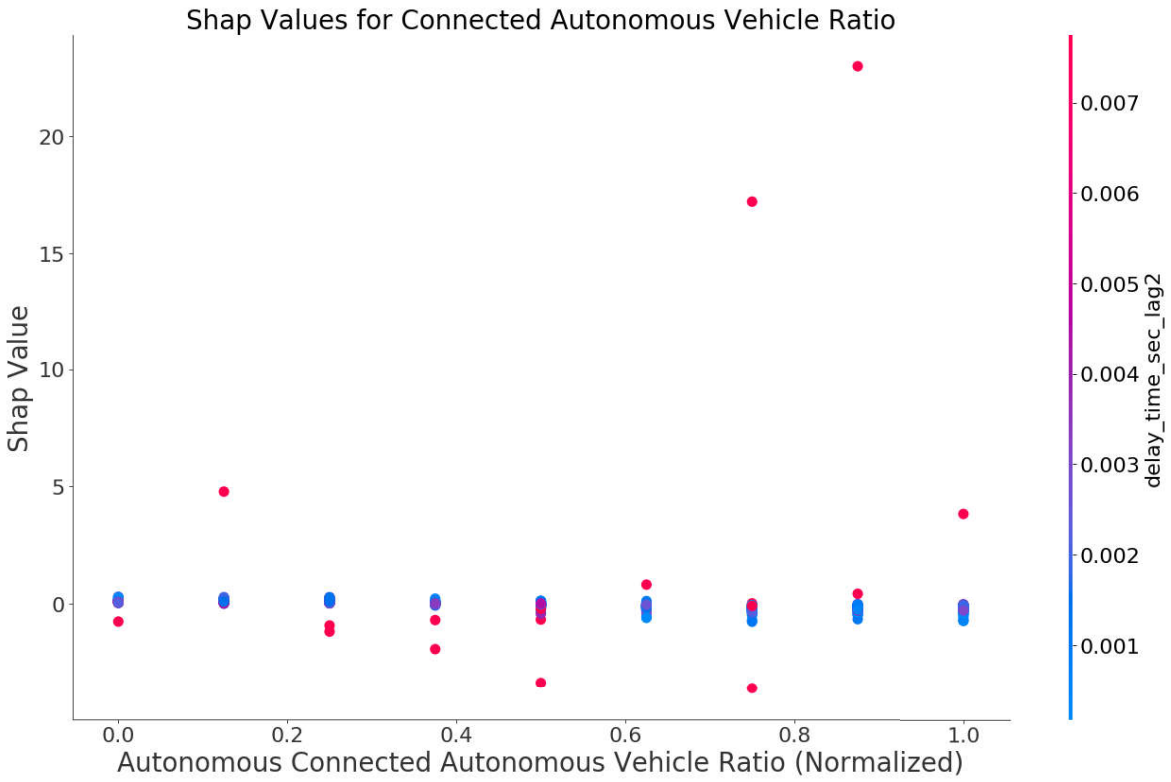

Figure 6.4. Shap Dependence Plot for CAV Ratio Feature of Random Forest Model.

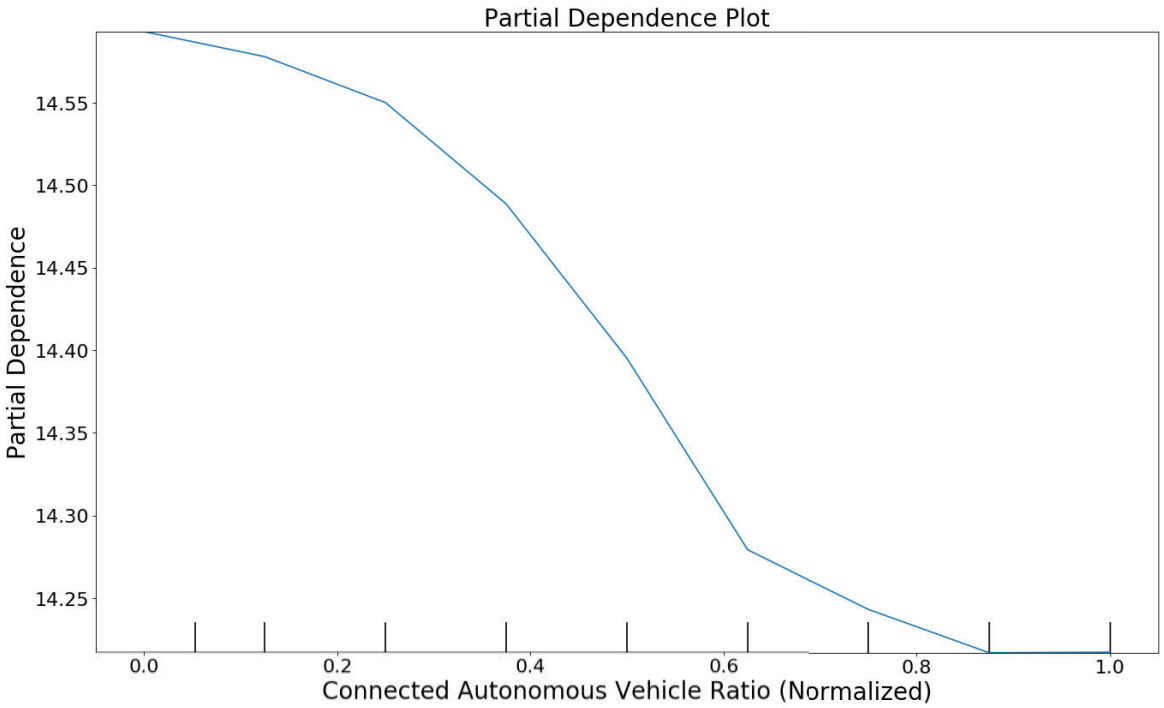

Figure 6.5. Partial Dependence Plot for CAV Ratio Feature of Random Forest Model.

This deduction can also be observed more clearly on a partial dependence plot. In Figure 6.5 it is seen that the average delay prediction (marginalized on all other features) made by the model decreases as the CAV ratio increases. Although the random forest model mainly learns from the other features due to their high cardinality and correlation to the target it still correctly learns the impact of CAVs on the delay.

6.2. Discussions

Since the overall validity of the model is discussed model's performance in various situations can be investigated.

The MSE of delay predictions and change in detector metrics with changing accident lane & connected vehicle ratio over all data can be found In Table 6.1. It is seen that the highest prediction errors are made when incident is on Lane 2 (middle lane). This difference between prediction errors for different accident lanes can be seen clearly in the Figure 6.6.

This is caused by the fact that accidents in Lane 2 (as it can be seen from the Figure 6.7, Figure 6.8 & Figure 6.9) create the highest and most variable delay values compared to the accidents in other lanes. It can be also seen that both delay and MSE generally decrease with the increase of the CAV ratio.

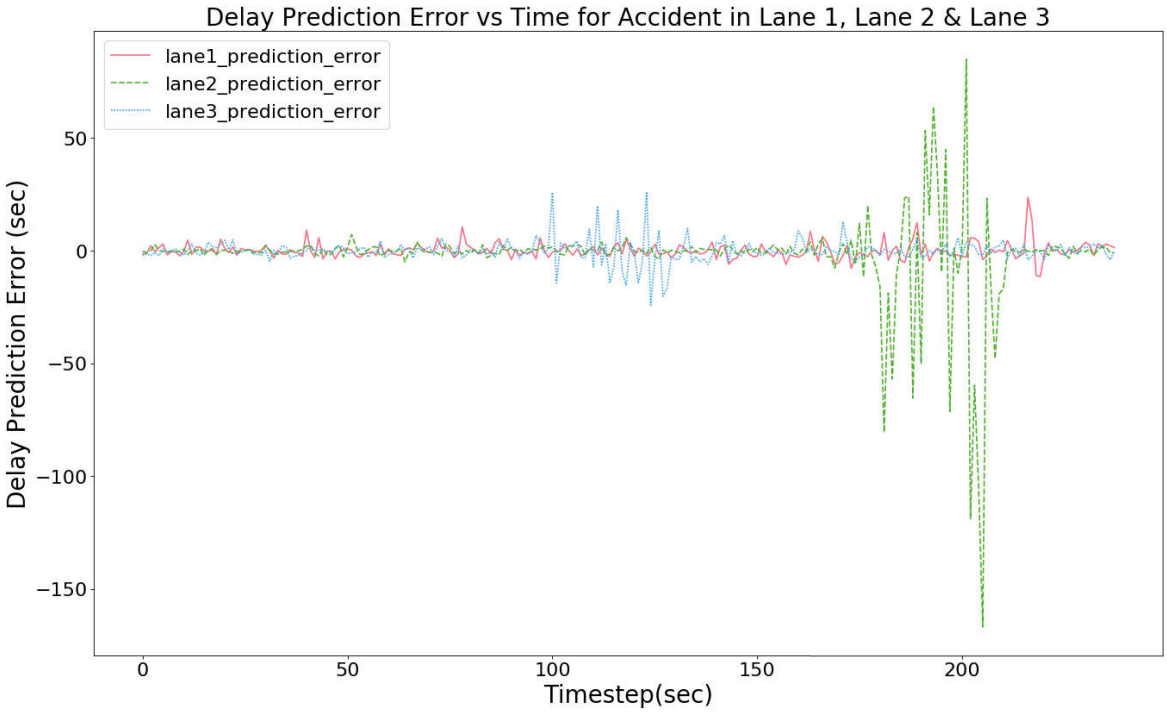

Figure 6.6. Delay Prediction Error Plot for Different Incident Lanes.

The prediction performance of the model on the accident local area (downstream section, incident section and upstream section) is analyzed in Table 6.2. For this analysis, accident happens at lane 1 of 7132 meter mark (detector 15) on 1227th second and lasts for 864 seconds. It is seen that model makes larger errors on incident section and makes smallest errors on downstream section. For the incident road section the delay values show more variance which results in amplified unexplainable noise. Also the model tries to generalize to the given training data and since most of the data points were coming from other road sections learning algorithm decreased the importance of the incident road section on the overall model. It is also noticed that as the CAV ratio inceases the delay values generally decrease with the MSE for the delay predictions.

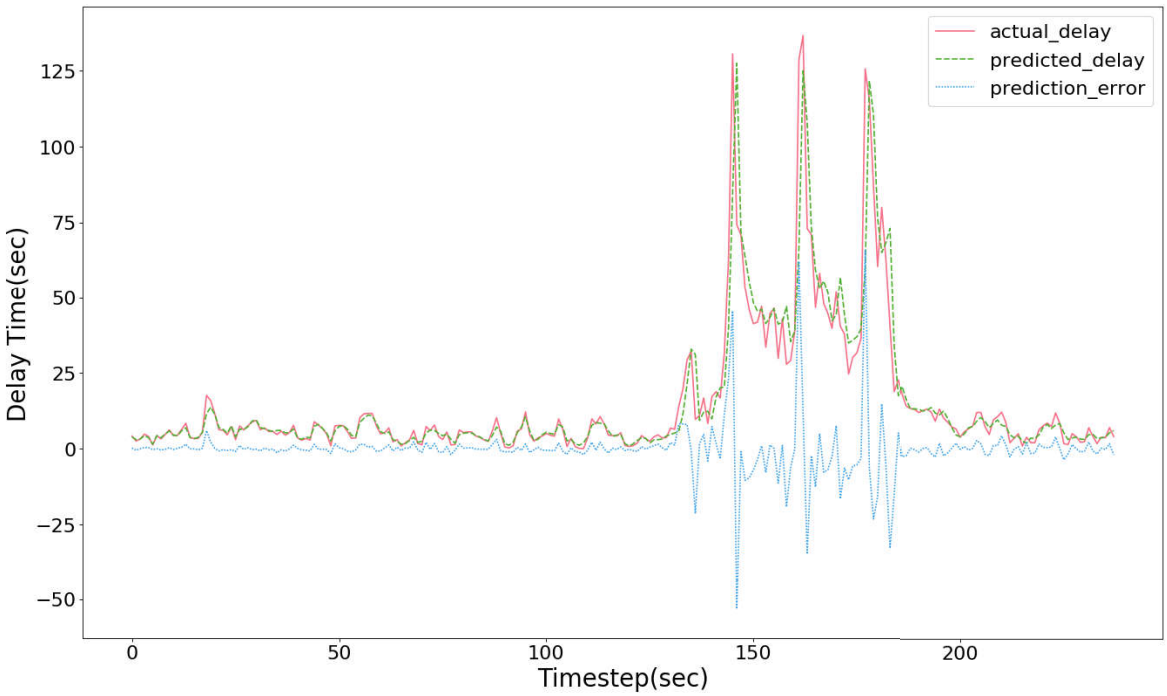

Figure 6.7. Delay Plot When Incident Is On Lane 1 (Rightmost Lane).

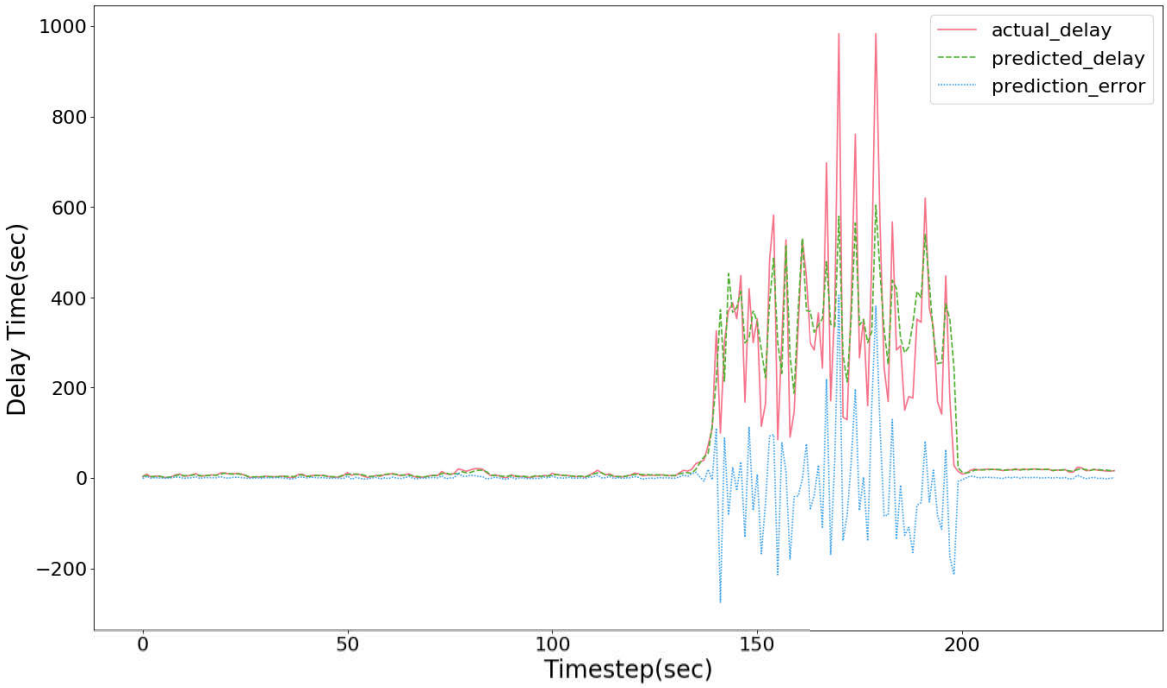

Figure 6.8. Delay Plot When Incident Is On Lane 2 (Middle Lane).

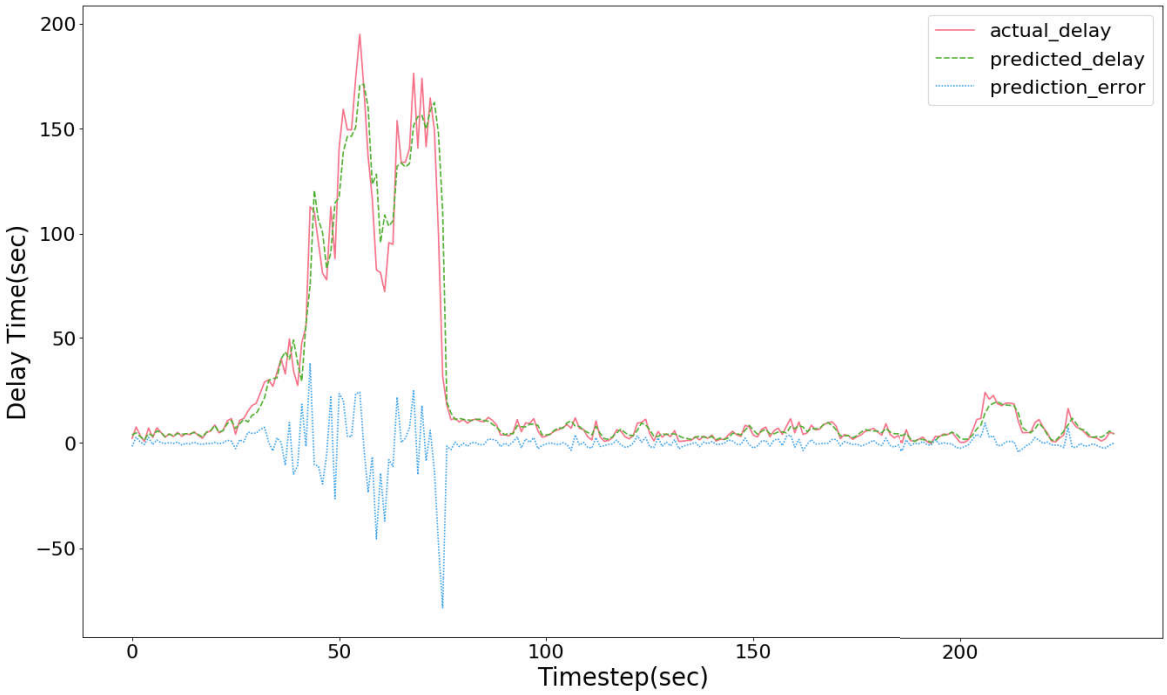

Figure 6.9. Delay Plot When Incident Is On Lane 3 (Leftmost Lane).

Table 6.1. Average Detector Metrics for Different Simulation Configurations

Incident Simulation	**Incident Lane**	**Connected Vehicle Ratio (%)**	**Flow (veh/h)**	**Change (%)**	**Density (veh/km)**	**Change (%)**	**Avg. Speed (km/h)**	**Change (%)**	**Section Travel Time (s)**	**Change (%)**	**Delay (s)**	**Change (%)**	**MSE**	**Change (%)**
FALSE	None	0	1409.14	0.00	11.62	0.00	67.36	0.00	32.92	0.00	15.68	0.00	185.63	0.00
TRUE	1	0	1394.71	0.00	11.29	0.00	66.53	0.00	29.37	0.00	12.13	0.00	30.90	0.00
		10	1403.40	0.62	11.03	-2.32	68.78	3.37	28.26	-3.75	11.02	-9.09	19.40	-37.22
		20	1399.21	0.32	10.54	-6.64	71.70	7.77	26.96	-8.18	9.72	-19.82	22.04	-28.68
		30	1401.88	0.51	10.14	-10.14	74.78	12.39	25.69	-12.51	8.45	-30.30	18.14	-41.31
		40	1403.86	0.66	9.77	-13.41	77.78	16.90	25.56	-12.98	8.32	-31.43	92.24	198.48
		50	1405.95	0.81	9.18	-18.66	81.32	22.22	23.18	-21.08	5.93	-51.06	13.77	-55.46
		60	1405.82	0.80	9.02	-20.12	83.71	25.82	22.61	-23.01	5.37	-55.72	13.73	-55.57
		70	1417.10	1.61	9.12	-19.20	85.26	28.14	22.55	-23.20	5.31	-56.20	20.46	-33.79
		80	1405.07	0.74	8.65	-23.35	87.87	32.07	21.57	-26.56	4.33	-64.32	15.21	-50.80
	2	0	1411.22	0.00	12.95	0.00	63.01	0.00	42.12	0.00	24.88	0.00	705.18	0.00
		10	1418.50	0.52	12.38	-4.39	65.83	4.47	39.16	-7.02	21.92	-11.89	625.07	-11.36
		20	1414.05	0.20	12.07	-6.79	67.87	7.71	38.30	-9.05	21.06	-15.33	687.01	-2.58
		30	1415.59	0.31	11.68	-9.82	70.76	12.30	36.80	-12.62	19.56	-21.37	650.38	-7.77
		40	1418.31	0.50	11.09	-14.35	74.31	17.94	34.72	-17.56	17.48	-29.72	608.55	-13.70
		50	1423.19	0.85	10.68	-17.53	77.59	23.14	33.44	-20.60	16.20	-34.88	883.99	25.36
		60	1420.10	0.63	10.06	-22.34	81.04	28.61	30.15	-28.40	12.91	-48.09	585.74	-16.94
		70	1429.93	1.33	10.27	-20.70	82.31	30.63	32.42	-23.02	15.18	-38.98	782.62	10.98
		80	1423.60	0.88	9.81	-24.25	84.98	34.86	30.60	-27.34	13.36	-46.28	955.42	35.49
	3	0	1397.22	0.00	11.79	0.00	64.99	0.00	32.20	0.00	14.96	0.00	44.72	0.00
		10	1407.62	0.74	11.50	-2.44	67.04	3.14	30.87	-4.12	13.63	-8.88	41.77	-6.61
		20	1405.01	0.56	11.00	-6.73	70.26	8.10	29.22	-9.24	11.98	-19.90	37.14	-16.95
		30	1404.27	0.50	10.78	-8.62	72.42	11.43	28.66	-10.98	11.42	-23.65	37.97	-15.11
		40	1408.58	0.81	10.06	-14.68	76.61	17.87	26.16	-18.75	8.92	-40.35	33.40	-25.32
		50	1410.18	0.93	9.73	-17.51	79.66	22.56	25.19	-21.78	7.94	-46.88	29.07	-35.00
		60	1410.68	0.96	9.40	-20.29	82.42	26.81	24.18	-24.89	6.94	-53.58	28.48	-36.33
		70	1413.13	1.14	9.26	-21.47	84.39	29.83	23.76	-26.20	6.52	-56.40	29.96	-33.02
		80	1404.77	0.54	8.86	-24.86	87.03	33.90	22.69	-29.51	5.45	-63.54	28.28	-36.76

Table 6.2. Detector Metrics on Incident Local Area for Different Connected Autonomous Vehicle Ratio Levels.

Connected Vehicle Ratio (%)	**Location**	**Flow (veh/h)**	**Change (%)**	**Density (veh/km)**	**Change (%)**	**Speed (km/h)**	**Change (%)**	**Section Travel Time (s)**	**Change (%)**	**Delay (s)**	**Change (%)**	**MSE**
0	Upstream	1493.78	0.00	15.14	0.00	59.58	0.00	37.15	0.00	19.91	0.00	64.47
	Incident	1566.05	0.00	17.48	0.00	59.25	0.00	37.65	0.00	20.41	0.00	217.06
	Donwstream	1404.03	0.00	10.73	0.00	69.73	0.00	26.96	0.00	9.72	0.00	9.36
10	Upstream	1450.76	-2.88	13.07	-13.65	65.35	9.69	33.94	-8.64	16.70	-16.12	55.11
	Incident	1529.08	-2.36	16.12	-7.77	62.82	6.01	35.89	-4.69	18.64	-8.66	214.99
	Donwstream	1390.92	-0.93	10.32	-3.77	71.36	2.34	26.61	-1.31	9.37	-3.63	10.16
20	Upstream	1437.31	-3.78	11.69	-22.81	70.44	18.24	28.74	-22.65	11.50	-42.26	21.64
	Incident	1535.46	-1.95	16.10	-7.91	63.69	7.48	34.93	-7.22	17.69	-13.33	152.72
	Donwstream	1398.99	-0.36	9.94	-7.38	74.71	7.14	25.24	-6.38	8.00	-17.70	12.52
30	Upstream	1397.65	-6.44	9.51	-37.18	76.37	28.19	24.70	-33.52	7.46	-62.54	13.60
	Incident	1512.27	-3.43	14.00	-19.92	70.78	19.45	28.20	-25.10	10.96	-46.30	52.12
	Donwstream	1402.69	-0.10	9.58	-10.73	77.45	11.07	24.01	-10.94	6.77	-30.35	8.47
40	Upstream	1397.98	-6.41	9.07	-40.11	80.88	35.75	22.84	-38.52	5.60	-71.87	5.73
	Incident	1450.42	-7.38	11.49	-34.28	74.72	26.10	26.60	-29.36	9.36	-54.16	47.82
	Donwstream	1406.05	0.14	8.97	-16.44	80.77	15.83	22.86	-15.22	5.62	-42.23	8.71
50	Upstream	1434.62	-3.96	12.32	-18.61	71.85	20.60	32.37	-12.86	15.13	-23.99	80.04
	Incident	1551.93	-0.90	16.49	-5.65	68.95	16.37	33.03	-12.28	15.79	-22.65	251.04
	Donwstream	1384.87	-1.36	8.91	-16.98	82.24	17.94	22.59	-16.21	5.35	-44.97	8.77
60	Upstream	1445.04	-3.26	11.05	-27.03	78.80	32.27	26.83	-27.77	9.59	-51.82	46.80
	Incident	1551.26	-0.94	15.81	-9.58	72.13	21.72	31.50	-16.34	14.26	-30.14	205.23
	Donwstream	1402.02	-0.14	8.77	-18.30	85.27	22.28	21.79	-19.17	4.55	-53.16	5.17
70	Upstream	1618.82	8.37	17.82	17.71	63.92	7.29	41.53	11.78	24.28	21.97	148.95
	Incident	1641.68	4.83	17.99	2.94	68.41	15.45	32.64	-13.31	15.40	-24.56	178.33
	Donwstream	1474.96	5.05	10.07	-6.18	82.53	18.35	23.49	-12.87	6.25	-35.71	7.32
80	Upstream	1399.66	-6.30	9.79	-35.33	81.71	37.15	24.31	-34.57	7.07	-64.50	7.52
	Incident	1447.06	-7.60	13.17	-24.64	73.60	24.21	28.85	-23.38	11.61	-43.13	142.56
	Donwstream	1366.39	-2.68	9.18	-14.41	84.56	21.27	23.13	-14.21	5.89	-39.43	5.23

Delay prediction plots for no-accident case and an accident case with accident on road section 14 can be found in Figure 6.10 and Figure 6.11 respectively. It can be seen from these figures that delay is lower for no-accident case and prediction errors are small and distributed around the mean of 0. For the accident case on the other hand the delay values make an increase after the accident starts. The predictions follow this trend however, cannot reach the peaks of the real data. It is also seen that the variance in the prediction error increases for the incident duration. Comparison of prediction errors throughout the simulation can be found in the Figure 6.12.

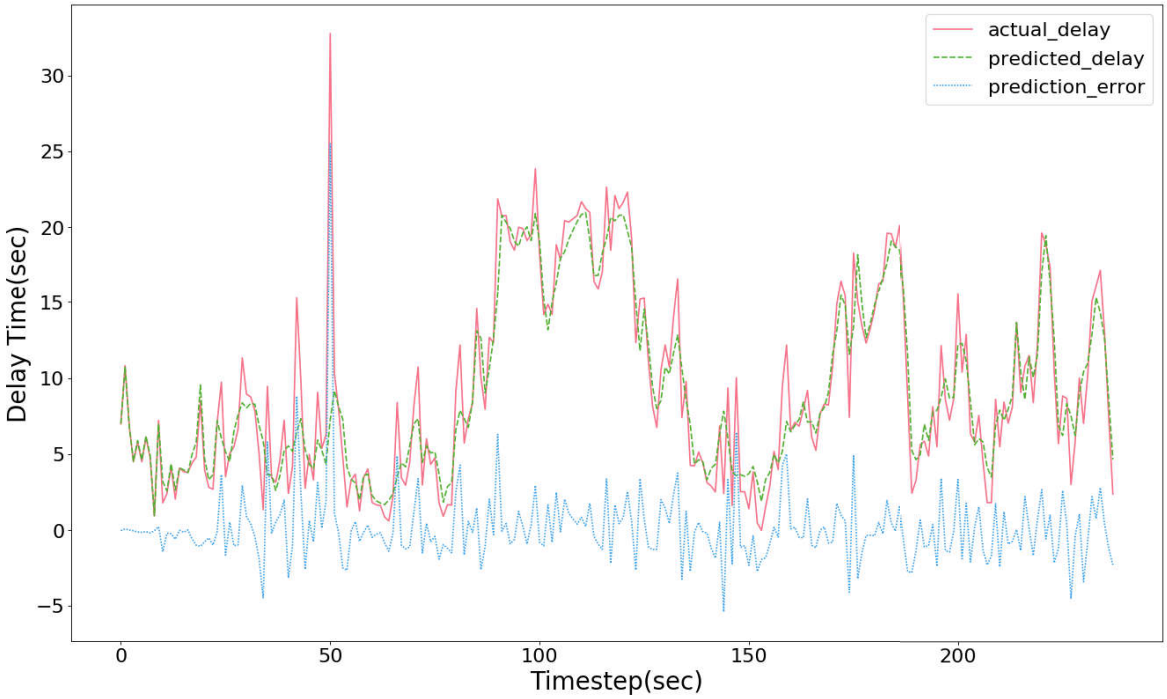

Figure 6.10. Delay Plot When There Is No Incident.

Delay prediction plots for the 0% CAV ratio and 80% CAV ratio are given in Figure 6.13 and Figure 6.14 respectively. It is observed that in the absence of CAVs after the incident starts the delay increases to a level and fluctuates around that increased delay level until the incident clears. However, when CAV ratio is increased to 80% the delay increase with the start of the incident and sharply decreases again and makes this reaction throughout the incident, creating a spiky graph. It is also seen that for the 80% CAV case the prediction errors are mostly negative for the incident duration.

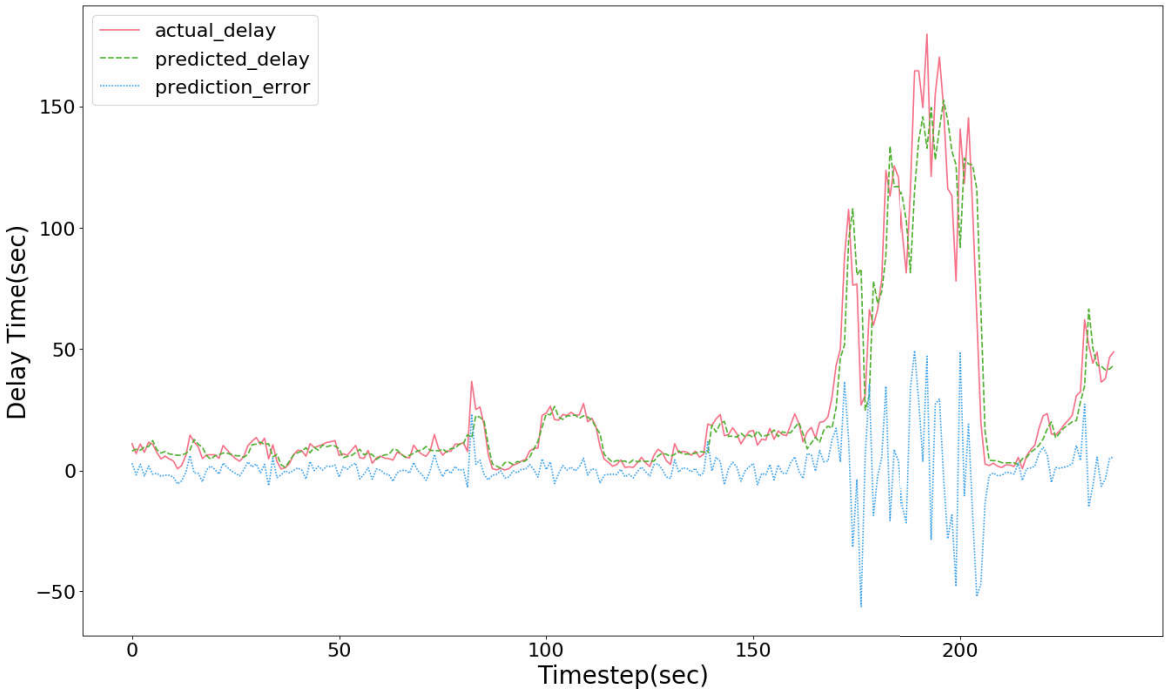

Figure 6.11. Delay Plot When There Is An Incident.

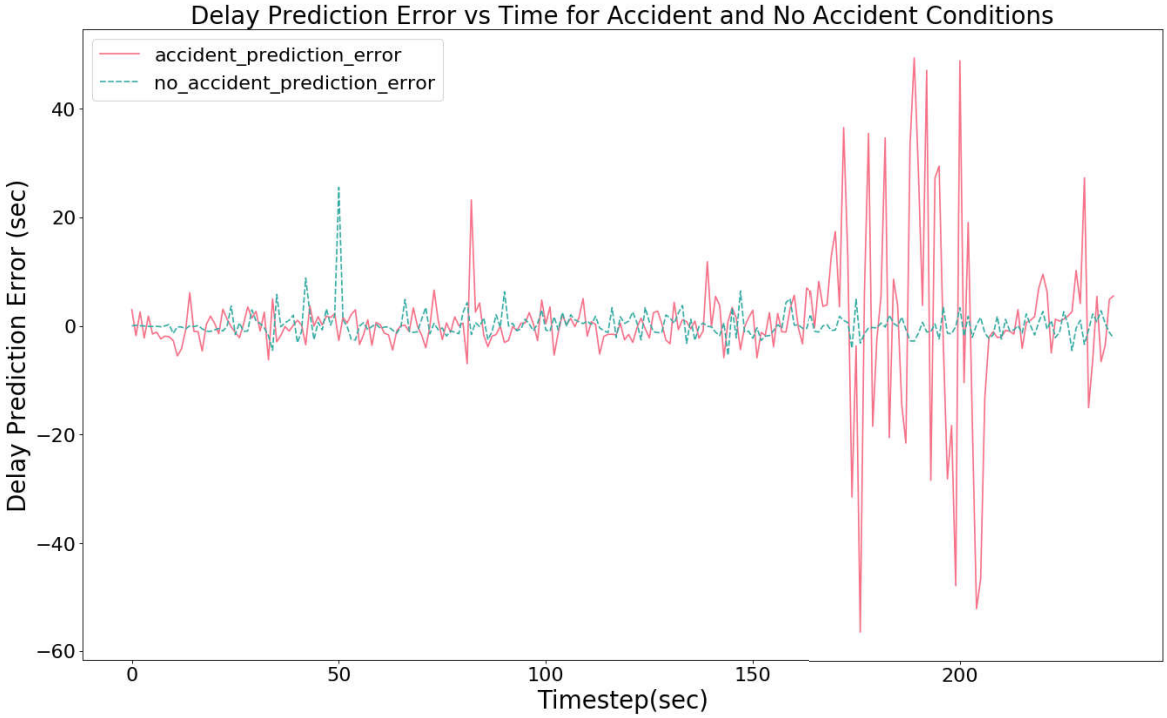

Figure 6.12. Delay Prediction Error Plot for No Accident & Accident Scenarios.

This indicates that the model overestimates the delay for higher CAV ratios which is caused by the fact that the model puts less importance on the CAV ratio then it should as explained in Section 6.1. Prediction errors for other CAV ratios can be found in Figure 6.15. Experienced delay and predicted delay plots for other CAV ratios can be found in the Appendix B.

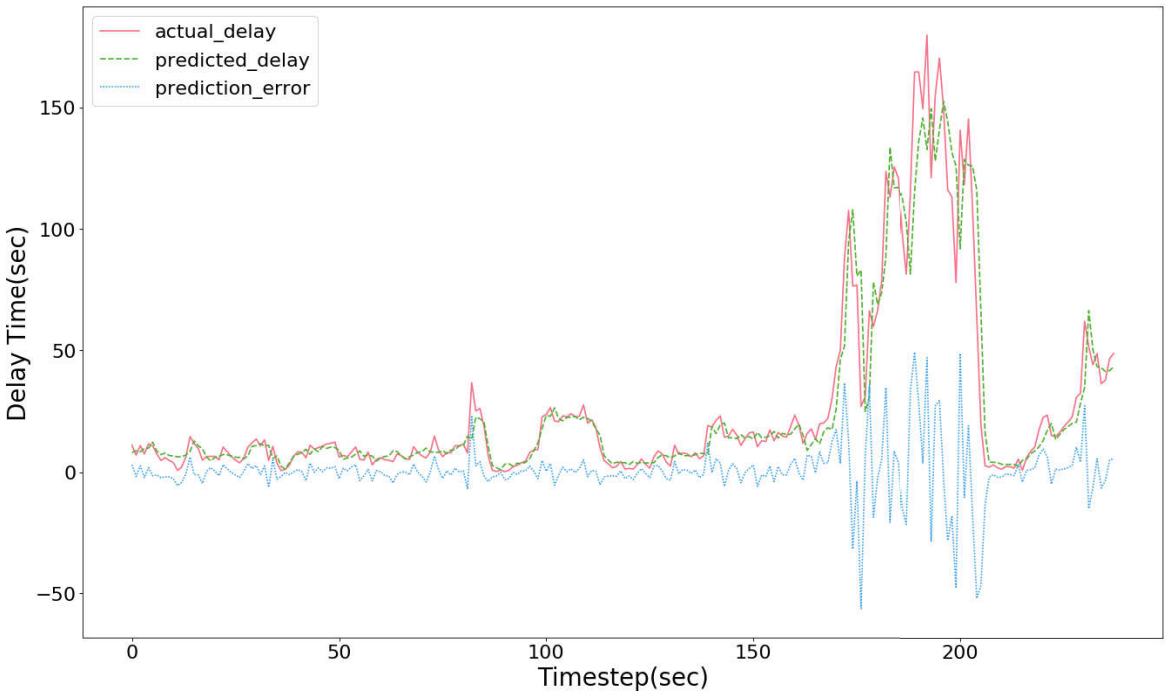

Figure 6.13. Delay Plot When Connected Autonomous Vehicle Ratio Is 0%.

The effect of incident duration can be discussed by inspecting the Figure 6.16 and Figure 6.17. For the long incident case the delay values are increased compared to the short incident case due to the accumulation of incoming demand. Prediction performance does not change significantly for these cases however, the high variance in the long duration case inevitably increase the prediction error.

The effect of CAV ratio on the network throughout the simulation time can be seen in the speed heatmaps of Figure 6.18 and Figure 6.19 which illustrates the traffic conditions for 20% and 40% CAV ratios. Increasing the CAV ratio regulate the overall traffic flow and decrease the shock wave effects of sudden lane changes or breaks made by human driven vehicles.

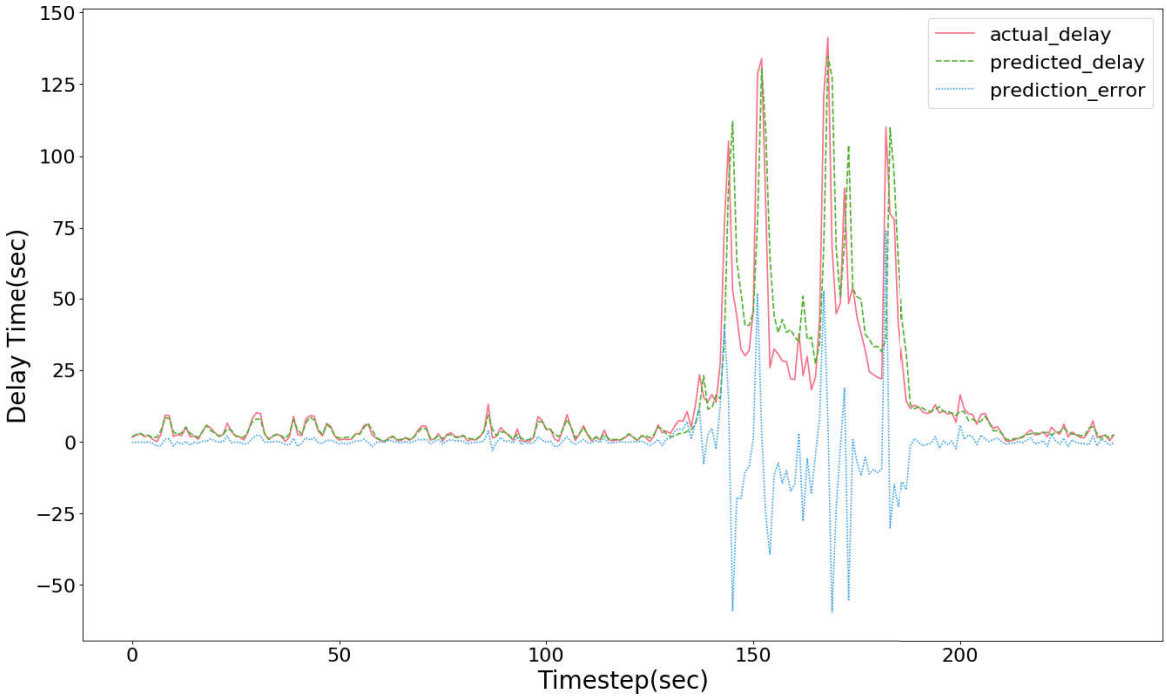

Figure 6.14. Delay Plot When Connected Autonomous Vehicle Ratio Is 80%.

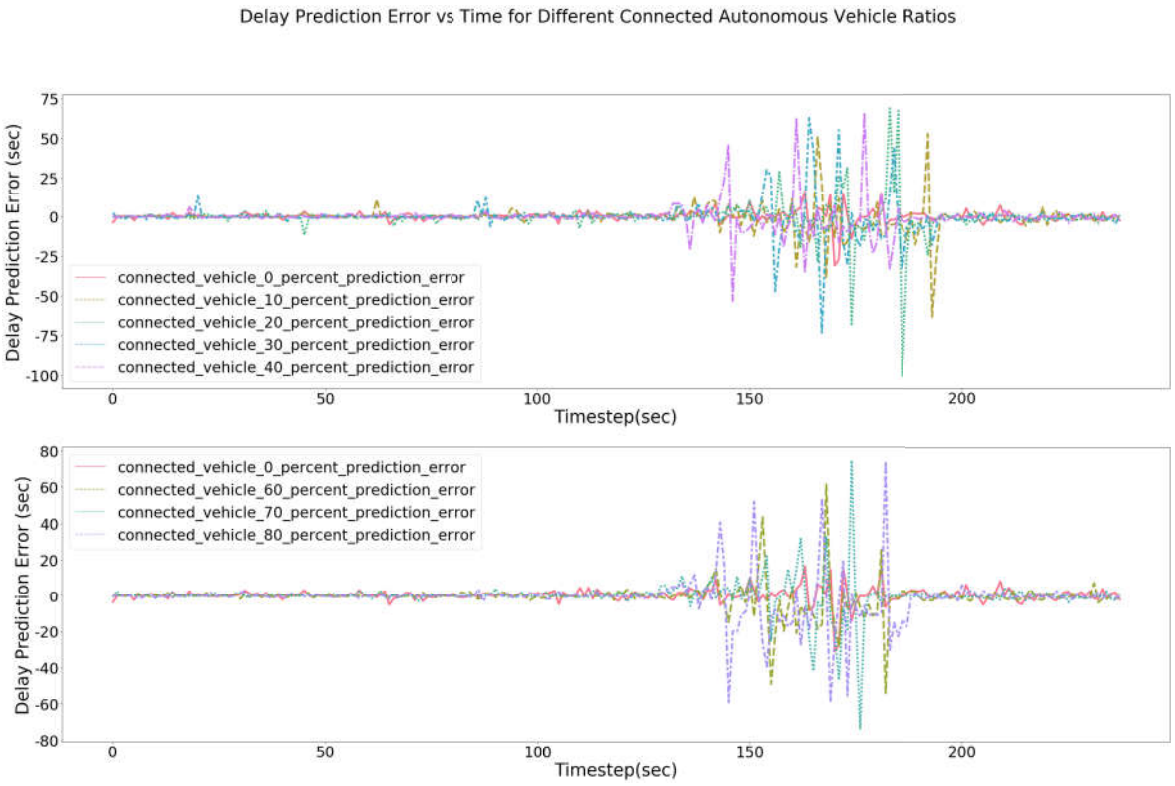

Figure 6.15. Delay Prediction Error Plot for Different CAV Ratio Levels.

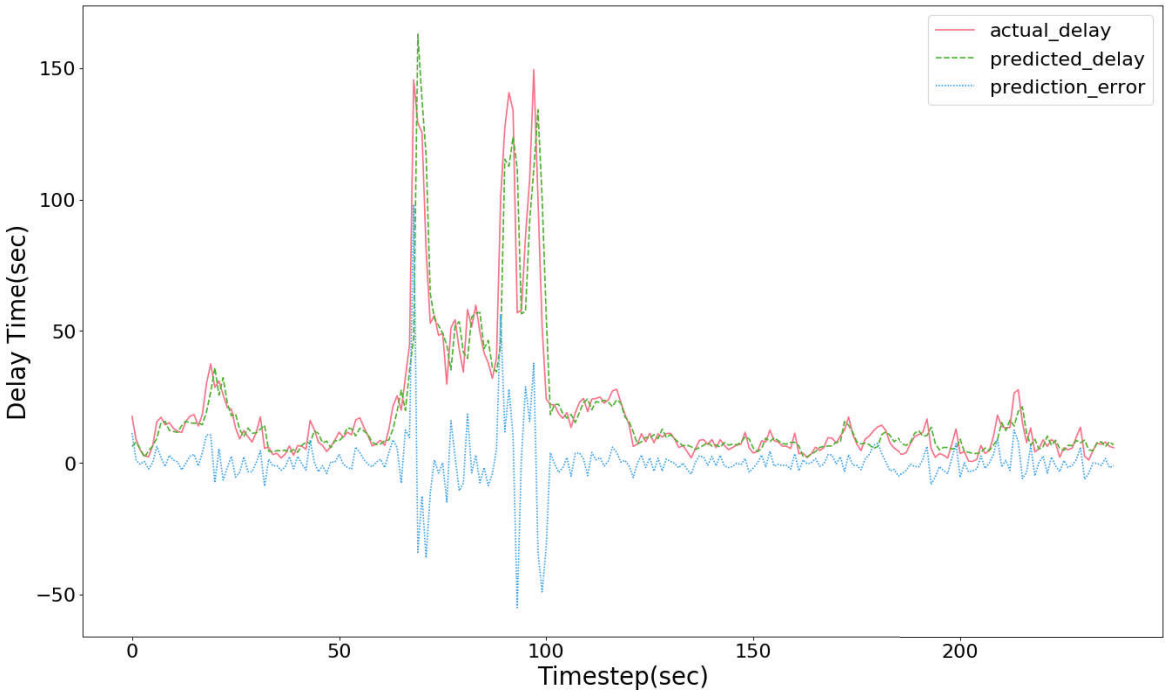

Figure 6.16. Delay Plot When Incident Duration Is Short (607 sec).

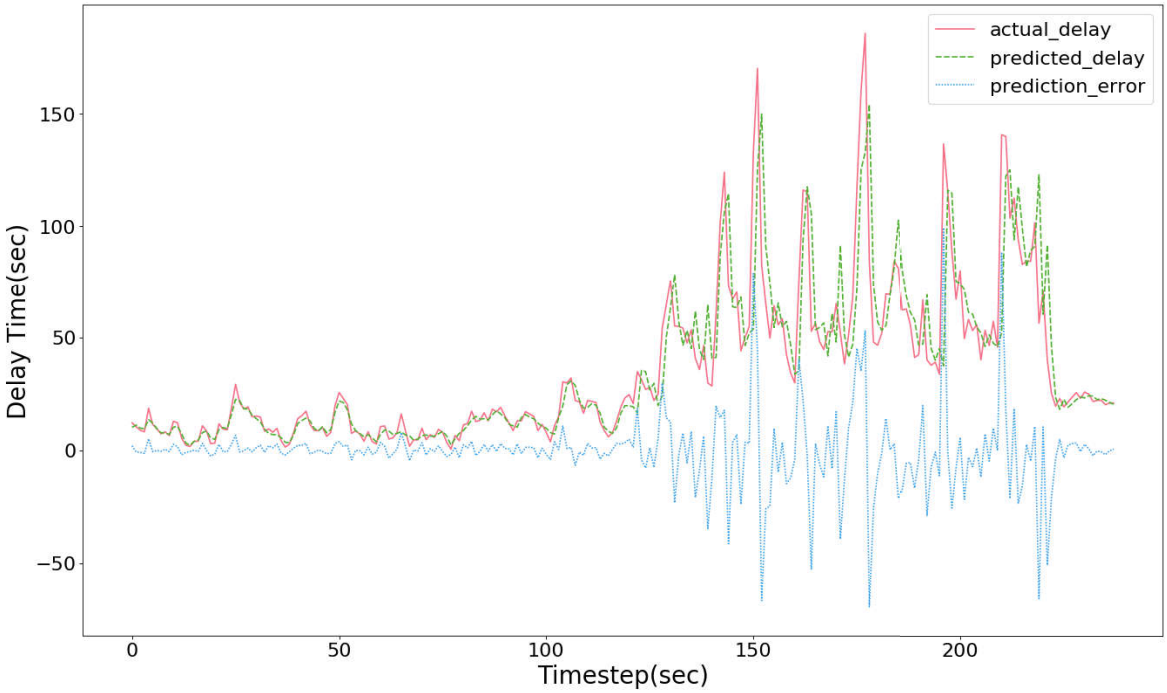

Figure 6.17. Delay Plot When Incident Duration Is Long (1492 sec).

The dark shaded triangle is thicker for the 20% CAV level meaning that the shock wave reaches to the further downstream sections. Also when the incident ends the queue formed by the incident clears faster for the 40% CAV ratio. Plots for other accident lanes and CAV ratios can be found in the Appendix C.

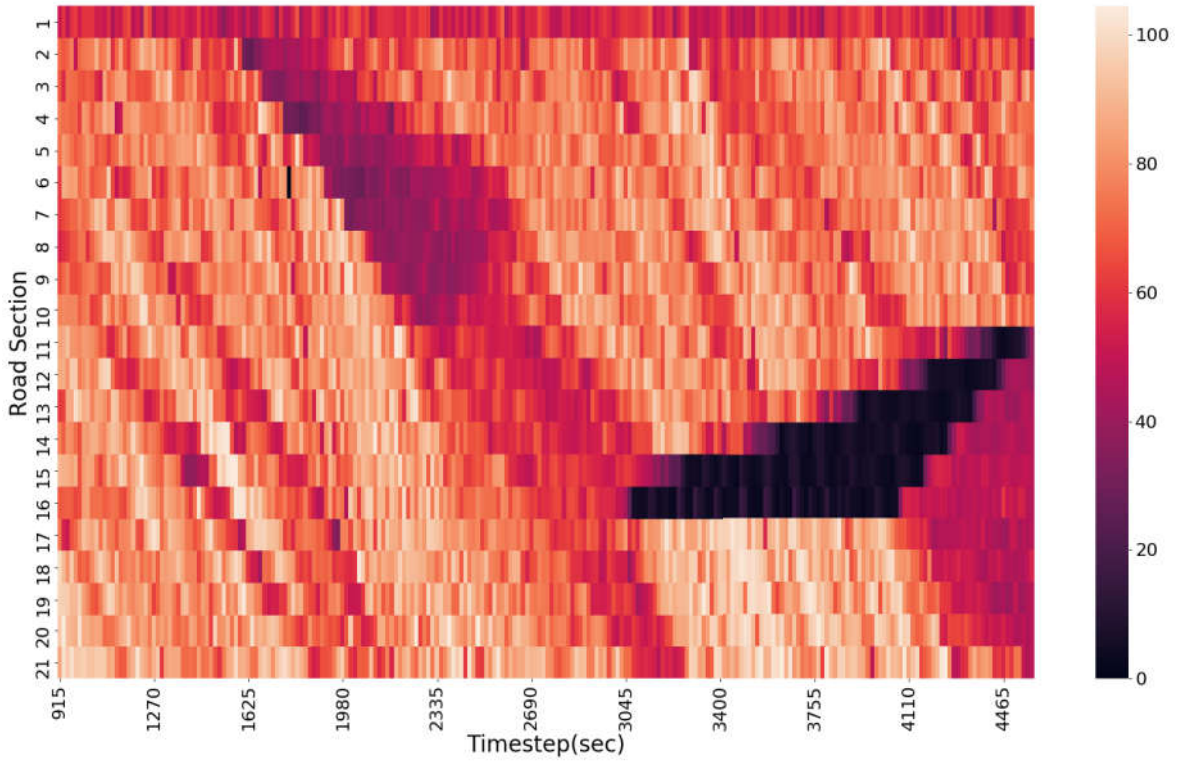

Figure 6.18. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 20%.

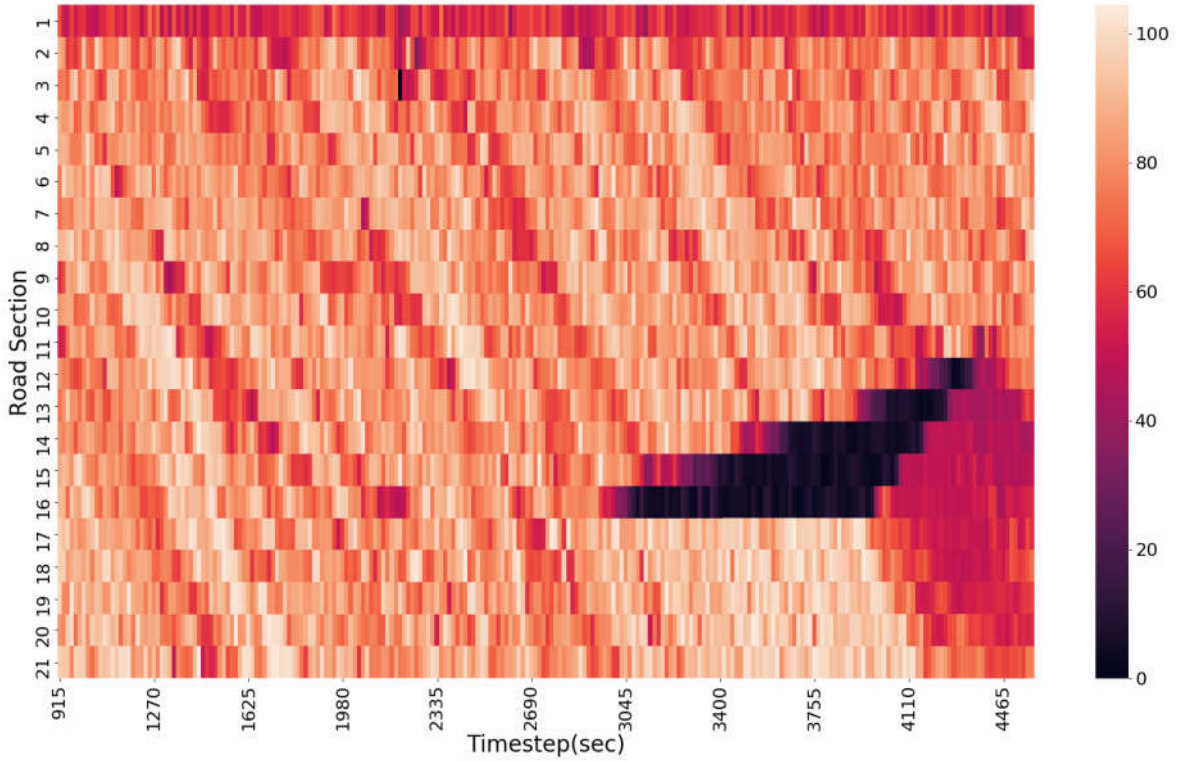

Figure 6.19. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 40%.

7. Conclusion

Traffic incidents create unwanted excess delay for the vehicles on the road. In this study a delay prediction model is built using the data generated by SUMO traffic simulation with different configurations.

After an exploratory data analysis feature engineering is conducted and incident related features, lagged detector features and adjacent detector features are produced. Enriched data is split into training and test data sets with consideration of the simulation parameters. The generated features are reduced using a feature selection methodology that use correlation coefficient, mutual information criterion and regular & standard random forest feature importance values. A variety of machine learning models are trained and tuned using cross validation with the selected features in training data set. After reviewing the cross validation results random forest is chosen to be the best performing model.

It is seen from the data that increasing the ratio of connected and autonomous vehicles in the simulation reduced the experienced delay. It is also seen from the SHAP values and partial dependence plot that the connected autonomous vehicle ratio also affects the predictions of the trained random forest model. However, the contribution of connected autonomous vehicle ratio in the trained model's predictions is smaller than it should have been. This is deduced by the fact that the model overestimates the delay for high connected autonomous vehicle ratios. The model might have corrected this overestimation if that feature was to be found more important during the learning process.

Since the simulation data had a fixed demand level and a fixed network type the estimations and findings of these study is only applicable for an uninterrupted flow with moderate demand level. Also, due to the limitations of random forest, model trained on this study cannot extrapolate outside its training data range.

In the literature usage of adjacent detector features with lagged features is not found to be prevalent however, these features are found to be quite informative for the given task. This methodology can also be used in the estimation of other metrics such as flow, density or speed for the next timesteps.

In this study only the previous and next detectors are taken as the adjacent detectors, however, in further studies more adjacent detectors can be incorporated into the feature set. The delay for the incident section is modelled together with the other sections however, it is seen that this approach deteriorated the model's predictions for the incident road section by lowering the delay predictions at peak delay values. In further studies incident section can be modelled separately from other road sections, loss function can be adjusted to give higher loss values for the prediction errors on the incident road section or new features that enable the model to separate the incident road section from other sections can be added.

REFERENCES

1. Khattak, A., X. Wang and H. Zhang, "Incident Management Integration Tool: Dynamically Predicting Incident Durations, Secondary Incident Occurrence and Incident Delays", *IET Intelligent Transport Systems*, Vol. 6, No. 2, pp. 204–214, 2012.

2. Islam, M. A., "A Literature Review on Freeway Traffic Incidents and Their Impact on Traffic Operations", *Journal of Transportation Technologies*, Vol. 9, No. 4, pp. 504–516, 2019.

3. Weng, J. and Q. Meng, "Estimating Capacity and Traffic Delay in Work Zones: An Overview", *Transportation Research Part C: Emerging Technologies*, Vol. 35, pp. 34–45, 2013.

4. Ziouva, E. and T. Antonakopoulos, "CSMA/CA Performance Under High Traffic Conditions: Throughput and Delay Analysis", *Computer Communications*, Vol. 25, No. 3, pp. 313–321, 2002.

5. Thiagarajan, A., L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo and J. Eriksson, "Vtrack: Accurate, Energy-aware Road Traffic Delay Estimation Using Mobile Phones", *In Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 85–98, 2009.

6. Ruan, L., M. P. I. Dias and E. Wong, "SmartBAN With Periodic Monitoring Traffic: A Performance Study on Low Delay and High Energy Efficiency", *IEEE Journal of Biomedical and Health Informatics*, Vol. 22, No. 2, pp. 471–482, 2016.

7. Minh, C. C., T. H. Binh, T. T. Mai and K. Sano, "The Delay Estimation Under Heterogeneous Traffic Conditions", *Journal of the Eastern Asia Society for Transportation Studies*, Vol. 8, pp. 1583–1595, 2010.

8. Liu, H., L. Rai, J. Wang and C. Ren, "A New Approach For Real-Time Traffic Delay Estimation Based on Cooperative Vehicle-Infrastructure Systems at the Signal Intersection", *Arabian Journal for Science and Engineering*, Vol. 44, No. 3, pp. 2613–2625, 2019.

9. Baykal-Gursoy, M., W. Xiao, Z. Duan and K. Ozbay, "Delay Estimation for Traffic Flow Interrupted by Incidents", *86th Annual Transportation Research Conf., Transportation Research Board, Washington, DC*, Citeseer, 2006.

10. Ban, X., R. Herring, P. Hao and A. M. Bayen, "Delay Pattern Estimation for Signalized Intersections Using Sampled Travel Times", Vol. 2130, No. 1, pp. 109–119, 2009.

11. Yang, C. D. and D. L. Fisher, "Safety Impacts and Benefits of Connected and Automated Vehicles: How Real Are They?", p. 1, 2021, `https://www.tandfonline.com/doi/full/10.1080/15472450.2021.1872143`, accessed in July 2021.

12. Guériau, M. and I. Dusparic, "Quantifying the Impact of Connected and Autonomous Vehicles on Traffic Efficiency and Safety in Mixed Traffic", *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–8, IEEE, 2020.

13. Al Islam, S. B., A. Hajbabaie and H. A. Aziz, "A Real-Time Network-Level Traffic Signal Control Methodology With Partial Connected Vehicle Information", *Transportation Research Part C: Emerging Technologies*, Vol. 121, p. 102830, 2020.

14. Guler, S. I., M. Menendez and L. Meier, "Using Connected Vehicle Technology to Improve the Efficiency of Intersections", *Transportation Research Part C: Emerging Technologies*, Vol. 46, pp. 121–131, 2014.

15. Hillier, J. A. and R. Rothery, "The Synchronization of Traffic Signals for Minimum

Delay", *Transportation Science*, Vol. 1, No. 2, pp. 81–94, 1967.

16. Li, X., A. J. Khattak and A. G. Kohls, "Signal Phase Timing Impact on Traffic Delay and Queue Length-a Intersection Case Study", *2016 Winter Simulation Conference (WSC)*, pp. 3722–3723, IEEE, 2016.

17. Mohajerpoor, R., M. Saberi and M. Ramezani, "Analytical Derivation of the Optimal Traffic Signal Timing: Minimizing Delay Variability and Spillback Probability for Undersaturated Intersections", *Transportation Research Part B: Methodological*, Vol. 119, pp. 45–68, 2019.

18. Al-Madani, H. M., "Dynamic Vehicular Delay Comparison Between a Police-Controlled Roundabout and a Traffic Signal", *Transportation Research Part A: Policy and Practice*, Vol. 37, No. 8, pp. 681–688, 2003.

19. Tian, Z. Z., T. Urbanik, R. Engelbrecht and K. Balke, "Variations in Capacity and Delay Estimates From Microscopic Traffic Simulation Models", *Transportation Research Record*, Vol. 1802, No. 1, pp. 23–31, 2002.

20. Chandra, S., A. Agrawal and A. Rajamma, "Microscopic Analysis of Service Delay At Uncontrolled Intersections in Mixed Traffic Conditions", *Journal of Transportation Engineering*, Vol. 135, No. 6, pp. 323–329, 2009.

21. Ngoduy, D., "Generalized Macroscopic Traffic Model With Time Delay", *Nonlinear Dynamics*, Vol. 77, No. 1, pp. 289–296, 2014.

22. Xi, J., W. Li, S. Wang and C. Wang, "An Approach to an Intersection Traffic Delay Study Based on Shift-Share Analysis", *Information*, Vol. 6, No. 2, pp. 246–257, 2015.

23. Lingras, P. and P. Mountford, "Time Delay Neural Networks Designed Using Genetic Algorithms for Short Term Inter-City Traffic Forecasting", *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent*

Systems, pp. 290–299, Springer, 2001.

24. Li, Y., C. Tang, S. Peeta and Y. Wang, "Nonlinear Consensus-Based Connected Vehicle Platoon Control Incorporating Car-Following Interactions and Heterogeneous Time Delays", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 6, pp. 2209–2219, 2018.

25. Tang, C. and Y. Li, "Consensus-Based Platoon Control For Non-Lane-Discipline Connected Autonomous Vehicles Considering Time Delays", *2018 37th Chinese Control Conference (CCC)*, pp. 7713–7718, IEEE, 2018.

26. Luan, Z., J. Zhang, W. Zhao and C. Wang, "Trajectory Tracking Control Of Autonomous Vehicle With Random Network Delay", *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 8, pp. 8140–8150, 2020.

27. Gou, G., Y. Zhao, J. Jiang and L. Shi, "The Prediction of Delay Time at Intersection and Route Planning For Autonomous Vehicles", p. 1, 2020, `https://ksiresearch.org/seke/seke20paper/paper018.pdf`, accessed in July 2021.

28. Song, X., X. Lou and L. Meng, "Time-Delay Feedback Cooperative Adaptive Cruise Control Of Connected Vehicles by Heterogeneous Channel Transmission", *Measurement and Control*, Vol. 52, No. 5-6, pp. 369–378, 2019.

29. Xu, S., H. Peng and Y. Tang, "Preview Path Tracking Control With Delay Compensation for Autonomous Vehicles", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 5, pp. 2979–2989, 2020.

30. Mehr, G. and A. Eskandarian, "Sentinel: An Onboard Lane Change Advisory System for Intelligent Vehicles to Reduce Traffic Delay During Freeway Incidents", *IEEE Transactions on Intelligent Transportation Systems*, 2021.

31. Galton, F., "Regression Towards Mediocrity in Hereditary Stature", *The Journal*

of the Anthropological Institute of Great Britain and Ireland, Vol. 15, pp. 246–263, 1886.

32. Ross, A., "Linear Regression", p. 13, 2018, `https://image.slidesharecdn.com/introductiontomachinelearningalgorithms-180312021825/95/introduction-to-machine-learning-algorithms-13-638.jpg`, accessed in July 2021.

33. Tipping, M. E., "Sparse Bayesian Learning and the Relevance Vector Machine", *Journal of Machine Learning Research*, Vol. 1, No. Jun, pp. 211–244, 2001.

34. Tikhonov, A. N. and V. Y. Arsenin, *Solutions of Ill-Posed Problems*, Vol. 1, Winston, New York, NY, USA, 1977.

35. Tibshirani, R., "Regression Shrinkage and Selection via the Lasso", *Journal of the Royal Statistical Society: Series B (Methodological)*, Vol. 58, No. 1, pp. 267–288, 1996.

36. Breiman, L., "Random Forests", *Machine learning*, Vol. 45, No. 1, pp. 5–32, 2001.

37. Wager, S., "Asymptotic Theory for Random Forests", *Arxiv Preprint Arxiv:1405.0352*, 2014.

38. Scornet, E., G. Biau and J.-P. Vert, "Consistency of Random Forests", *The Annals of Statistics*, Vol. 43, No. 4, pp. 1716–1741, 2015.

39. Hastie, T., R. Tibshirani and J. Friedman, "Random Forests", *The Elements of Statistical Learning*, pp. 587–604, Springer, 2009.

40. Chen, T. and C. Guestrin, "Xgboost: A Scalable Tree Boosting System", *Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

41. DenisBoigelot, "Several Sets of (X, Y) Points, With the Pearson Correlation Co-

efficient of X and Y for Each Set. The Correlation Reflects the Noisiness and Direction of a Linear Relationship (Top Row), but Not the Slope of That Relationship (Middle), nor Many Aspects of Nonlinear Relationships (Bottom)", p. 1, 2008, https://en.wikipedia.org/wiki/Correlation_and_dependence, accessed in July 2021.

42. Beraha, M., A. M. Metelli, M. Papini, A. Tirinzoni and M. Restelli, "Feature Selection via Mutual Information: New Theoretical Insights", *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, IEEE, 2019.

43. Battini, D., "XOR Problem", p. 1, 2019, https://www.tech-quantum.com/solving-xor- problem-using-neural-network-c, accessed in July 2021.

44. Deng, H. and G. Runger, "Feature Selection via Regularized Trees", *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2012.

45. de Sousa, M. N., R. Sant'Ana, R. P. Fernandes, J. C. Duarte, J. A. Aploinário and R. S. Thomä, "Comparison of Random Forest and Gradient Boosting Fingerprints to Enhance an Outdoor Radio-Frequency Localization System", p. 1, 2020.

46. Golden, C. E., M. J. Rothrock Jr and A. Mishra, "Comparison Between Random Forest and Gradient Boosting Machine Methods for Predicting Listeria Spp. Prevalence in the Environment of Pastured Poultry Farms", *Food Research International*, Vol. 122, pp. 47–55, 2019.

47. Lundberg, S. M. and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions", *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4768–4777, 2017.

APPENDIX A: PERMUTATION FEATURE IMPORTANCE

Table A.1: Permutation Feature Importance Values.

Feature Name	**Mean Decrease In MSE**
section_travel_time_sec	381.85
delay_time_sec	264.96
avg_speed_kmph	133.29
next_detector_delay_time_sec_lag5	42.01
next_detector_avg_speed_kmph_lag5	40.33
avg_speed_kmph_lag1	31.84
next_detector_section_travel_time_sec_lag5	31.71
delay_time_sec_lag1	30.62
next_detector_flow_vehph_lag4	30.05
section_travel_time_sec_lag1	27.86
density_vehpkm_lag1	21.94
density_vehpkm	21.38
next_detector_density_vehpkm_lag5	20.29
next_detector_avg_speed_kmph_lag6	19.54
next_detector_section_travel_time_sec_lag6	18.91
next_detector_density_vehpkm_lag8	16.79
next_detector_density_vehpkm_lag6	16.40
next_detector_delay_time_sec_lag6	16.16
next_detector_flow_vehph_lag8	15.35
next_detector_delay_time_sec_lag4	12.92
connected_vehicle_ratio	12.13
delay_time_sec_lag2	10.65
next_detector_section_travel_time_sec_lag4	10.51
section_travel_time_sec_lag3	10.41

Table A.1. Permutation Feature Importance Values. (cont.)

Feature Name	**Mean Decrease In MSE**
section_travel_time_sec_lag2	9.46
delay_time_sec_lag3	9.42
next_detector_section_travel_time_sec_lag1	9.23
next_detector_avg_speed_kmph_lag1	8.70
accident_lane_2	8.11
avg_speed_kmph_lag2	8.02
avg_speed_kmph_lag3	7.82
next_detector_section_travel_time_sec_lag7	7.11
prev_detector_density_vehpkm_lag3	6.95
density_vehpkm_lag2	6.38
prev_detector_avg_speed_kmph_lag5	6.30
next_detector_delay_time_sec_lag7	6.25
delay_time_sec_lag5	5.62
avg_speed_kmph_lag4	5.37
section_travel_time_sec_lag5	5.30
delay_time_sec_lag4	5.09
delay_time_sec_lag6	4.37
section_travel_time_sec_lag4	4.23
prev_detector_section_travel_time_sec_lag2	4.21
avg_speed_kmph_lag8	4.19
delay_time_sec_lag7	4.02
prev_detector_section_travel_time_sec_lag3	3.97
section_travel_time_sec_lag6	3.96
prev_detector_avg_speed_kmph_lag1	3.70
delay_time_sec_lag8	3.69
prev_detector_avg_speed_kmph_lag2	3.44
prev_detector_delay_time_sec_lag3	3.42
section_travel_time_sec_lag7	3.39

Table A.1. Permutation Feature Importance Values. (cont.)

Feature Name	**Mean Decrease In MSE**
section_travel_time_sec_lag8	3.39
distance_to_accident	3.39
prev_detector_delay_time_sec_lag2	3.39
prev_detector_avg_speed_kmph_lag3	3.38
prev_detector_section_travel_time_sec_lag1	3.22
prev_detector_delay_time_sec_lag1	3.11
prev_detector_delay_time_sec_lag4	3.05
prev_detector_avg_speed_kmph_lag4	2.99
time_after_accident_started	2.80
prev_detector_section_travel_time_sec_lag4	2.67
accident_lane_3	1.39
is_accident_timestamp	0.27
accident_lane_1	0.07

APPENDIX B: DELAY PLOTS FOR DIFFERENT CONNECTED AUTONOMOUS VEHICLE LEVELS

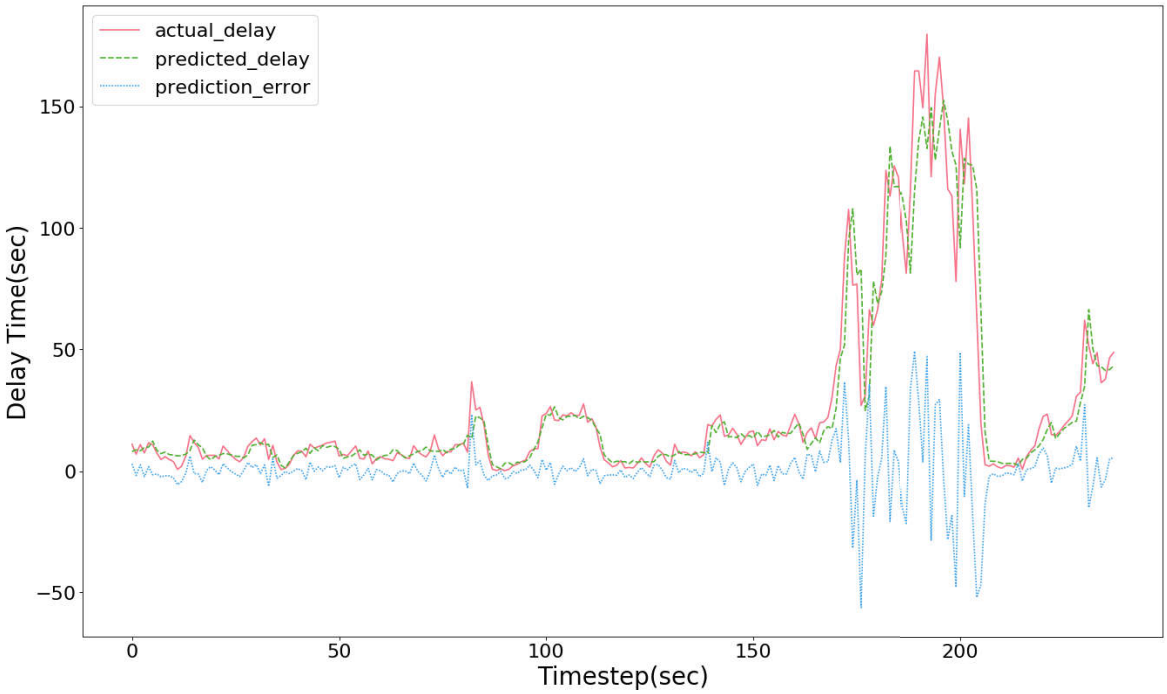

Figure B.1. Delay Plot When Connected Autonomous Vehicle Ratio Is 0%.

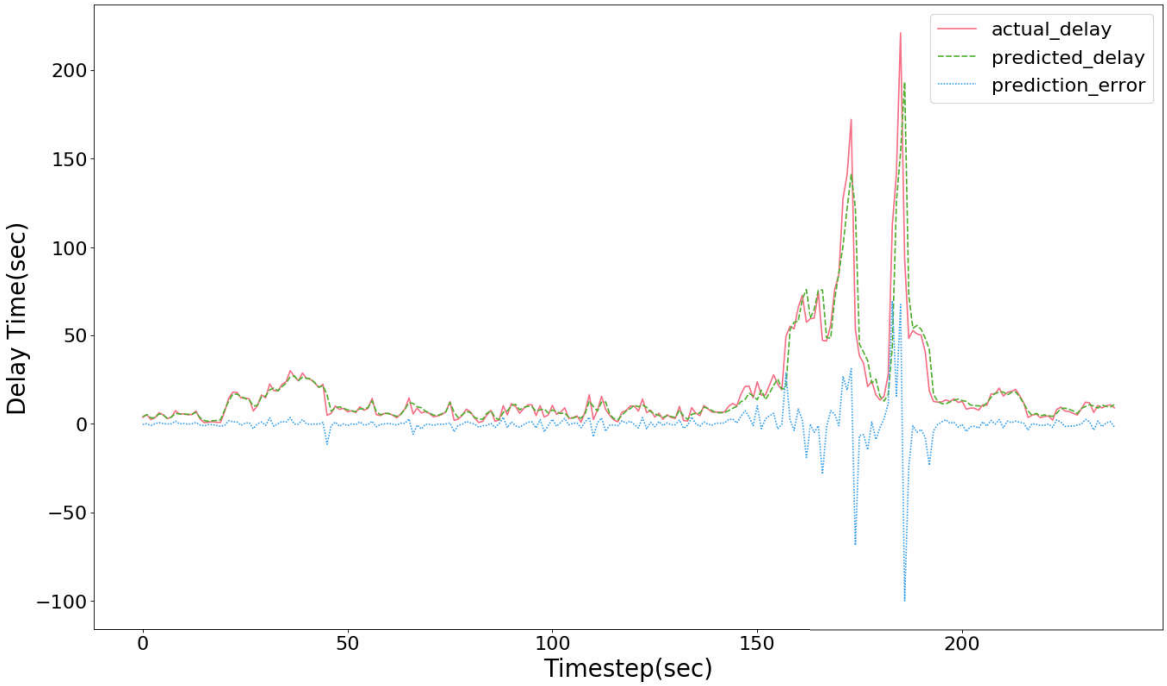

Figure B.2. Delay Plot When Connected Autonomous Vehicle Ratio Is 20%.

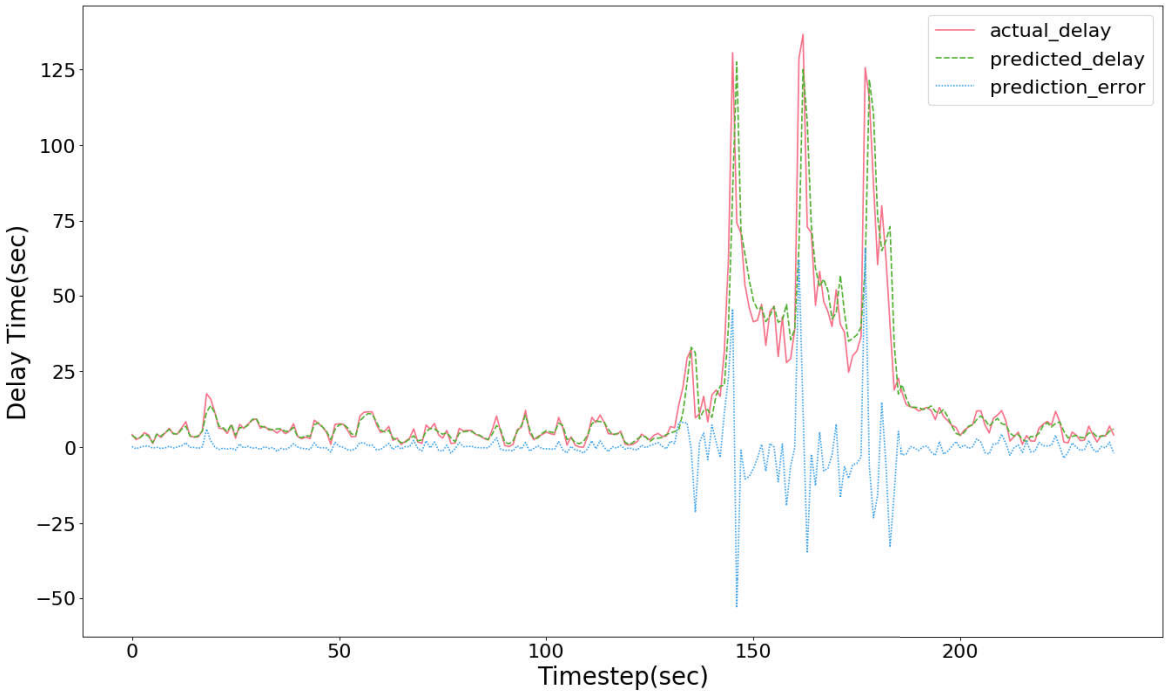

Figure B.3. Delay Plot When Connected Autonomous Vehicle Ratio Is 40%.

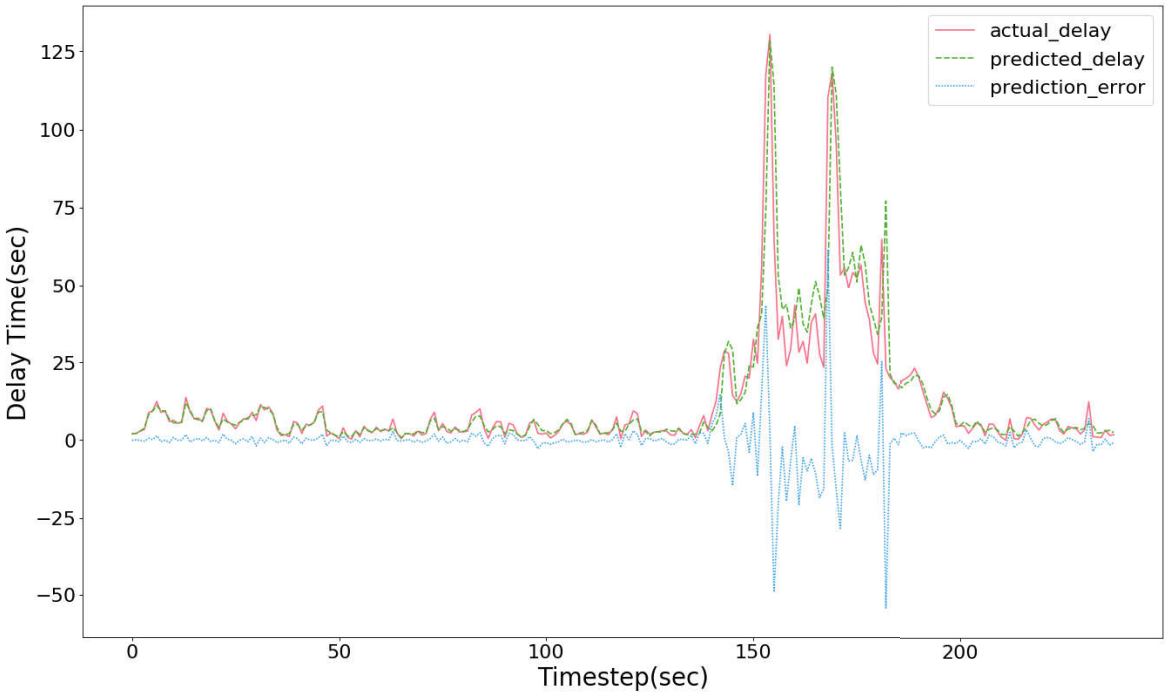

Figure B.4. Delay Plot When Connected Autonomous Vehicle Ratio Is 60%.

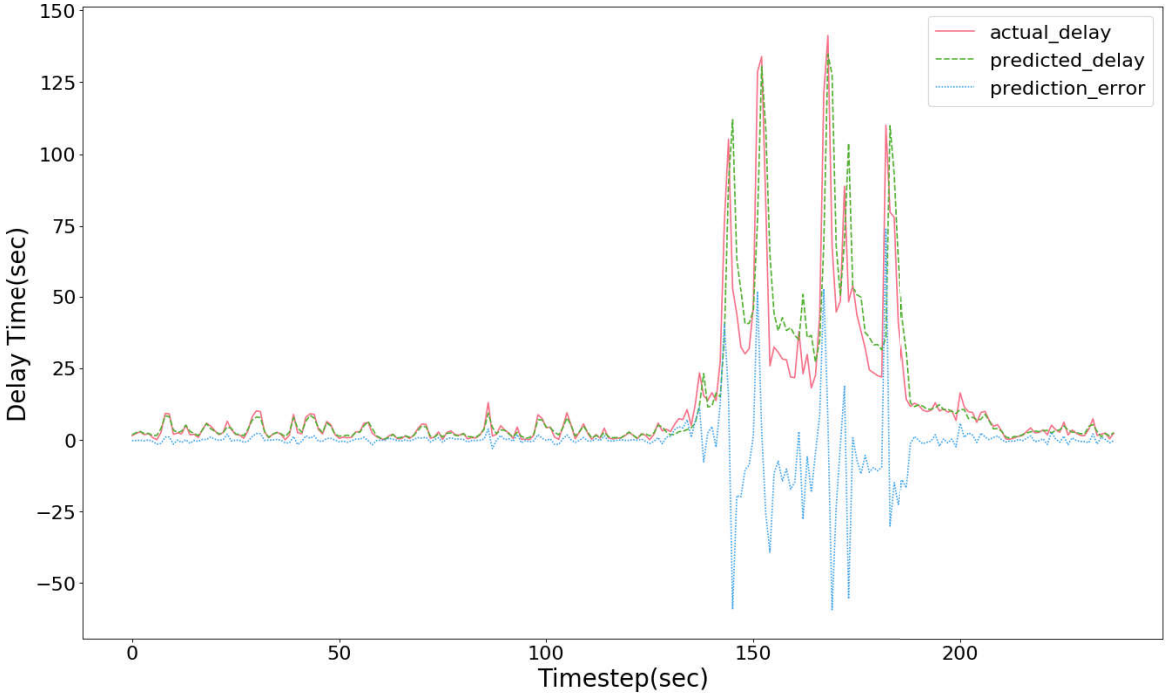

Figure B.5. Delay Plot When Connected Autonomous Vehicle Ratio Is 80%.

APPENDIX C: SPEED HEATMAP FOR DIFFERENT ACCIDENT LANES & CONNECTED AUTONOMOUS VEHICLE LEVELS

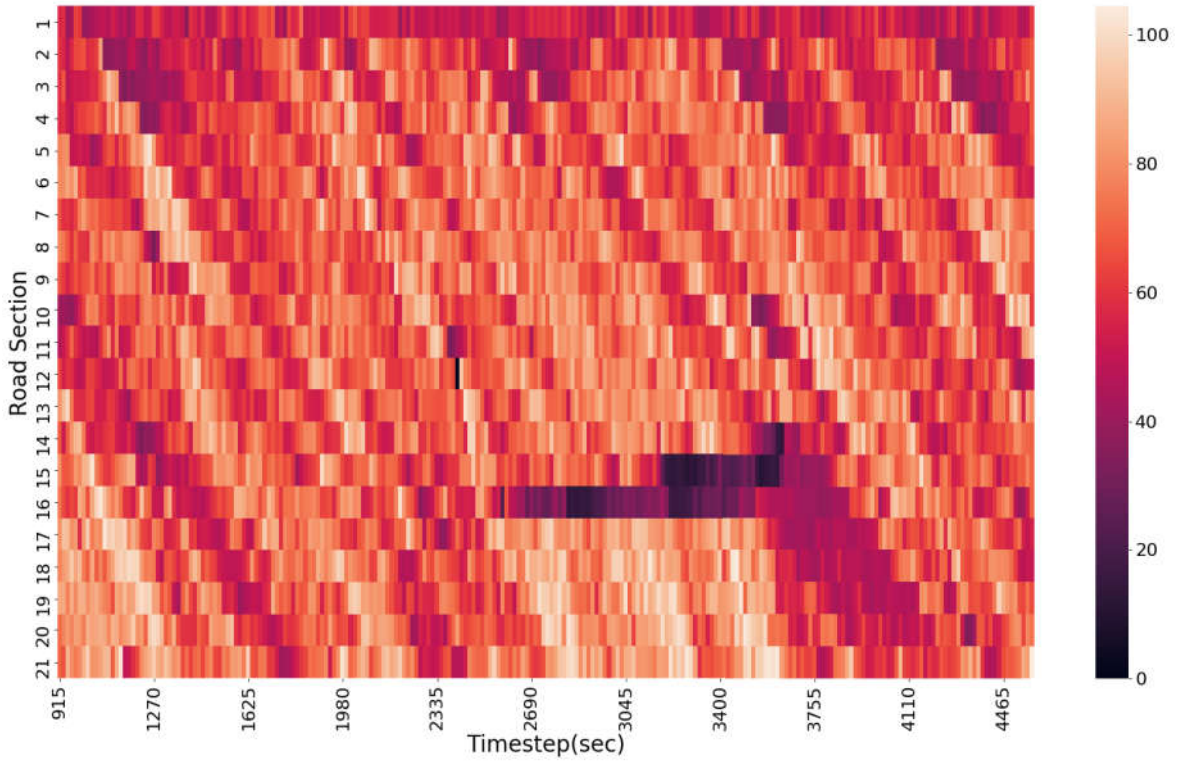

Figure C.1. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 0%.

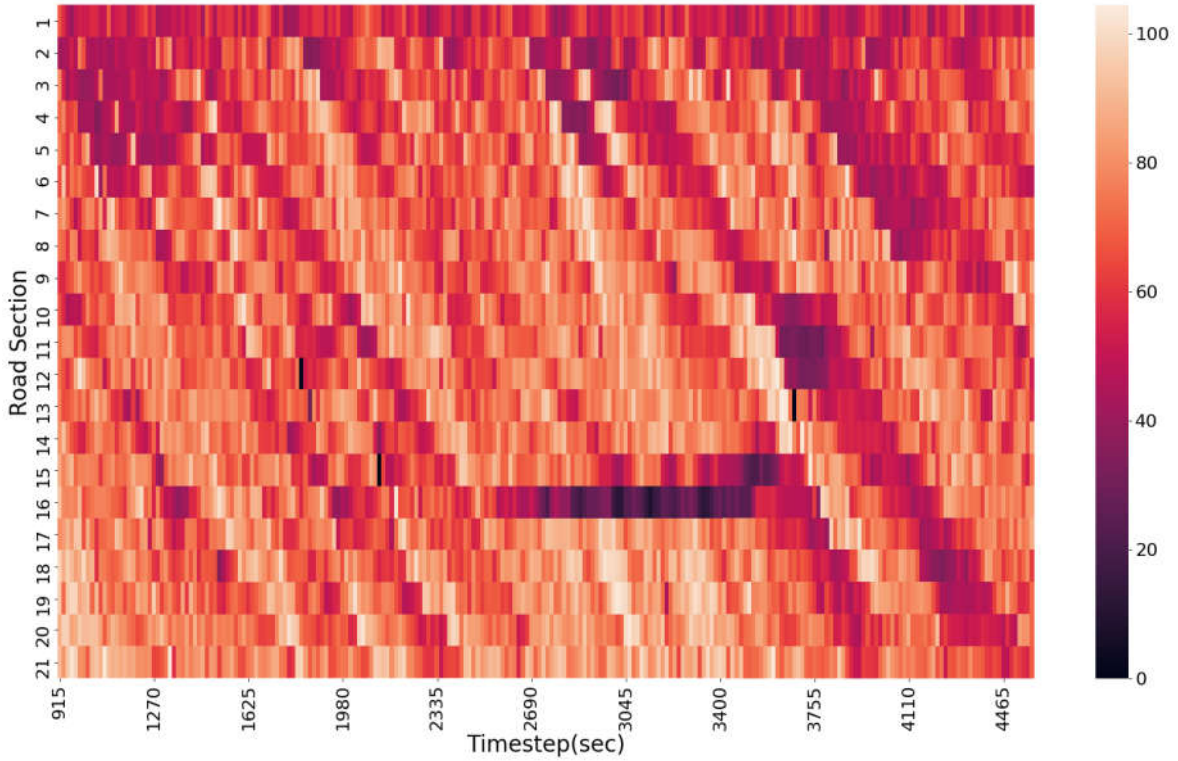

Figure C.2. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 10%.

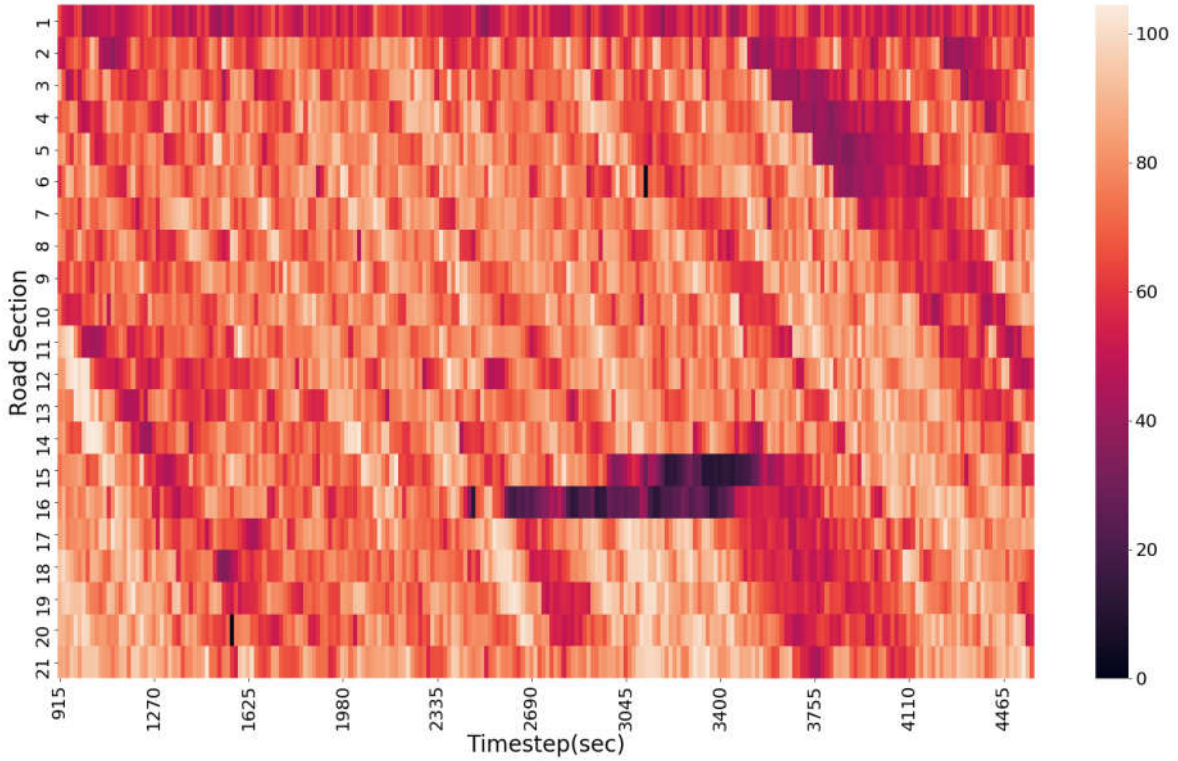

Figure C.3. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 20%.

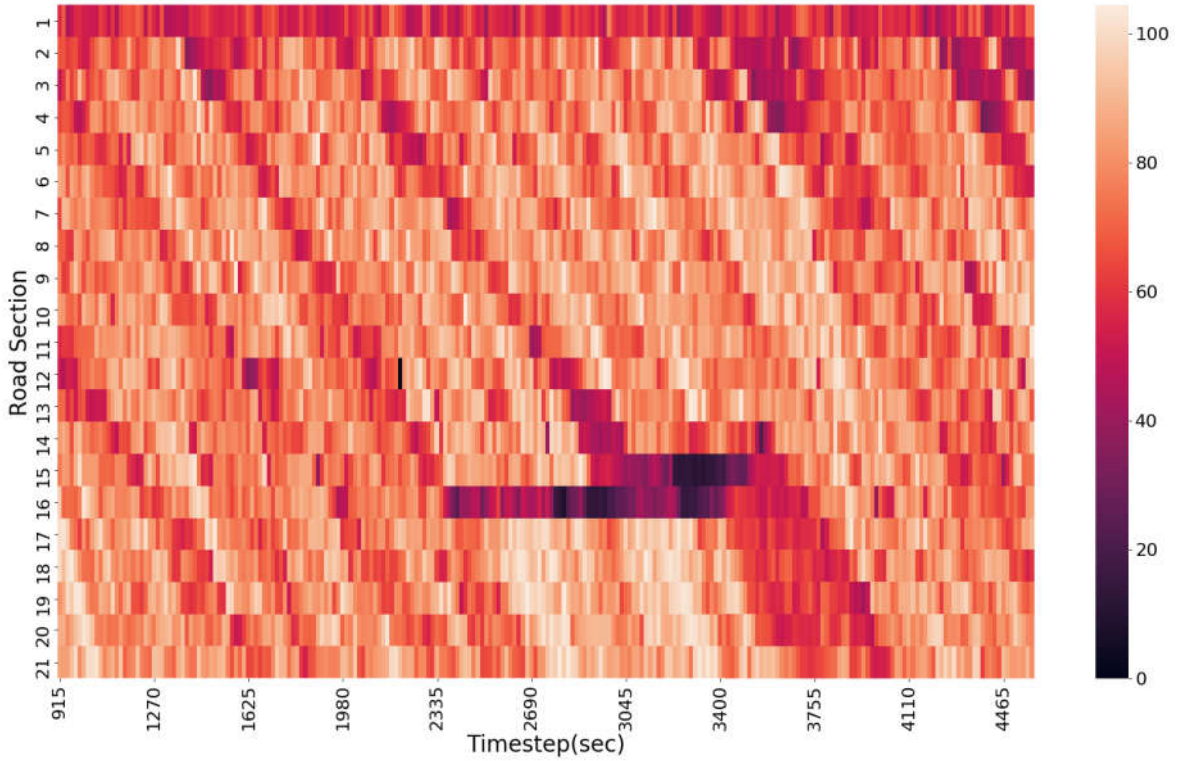

Figure C.4. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 30%.

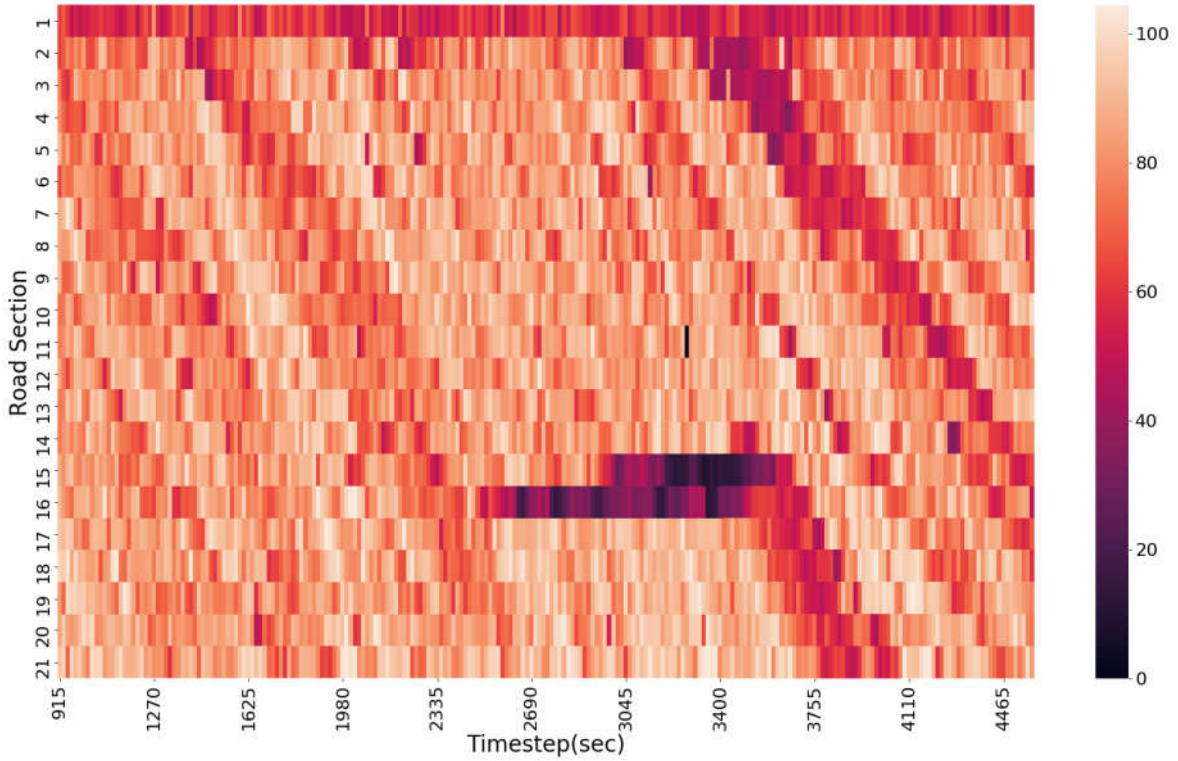

Figure C.5. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 40%.

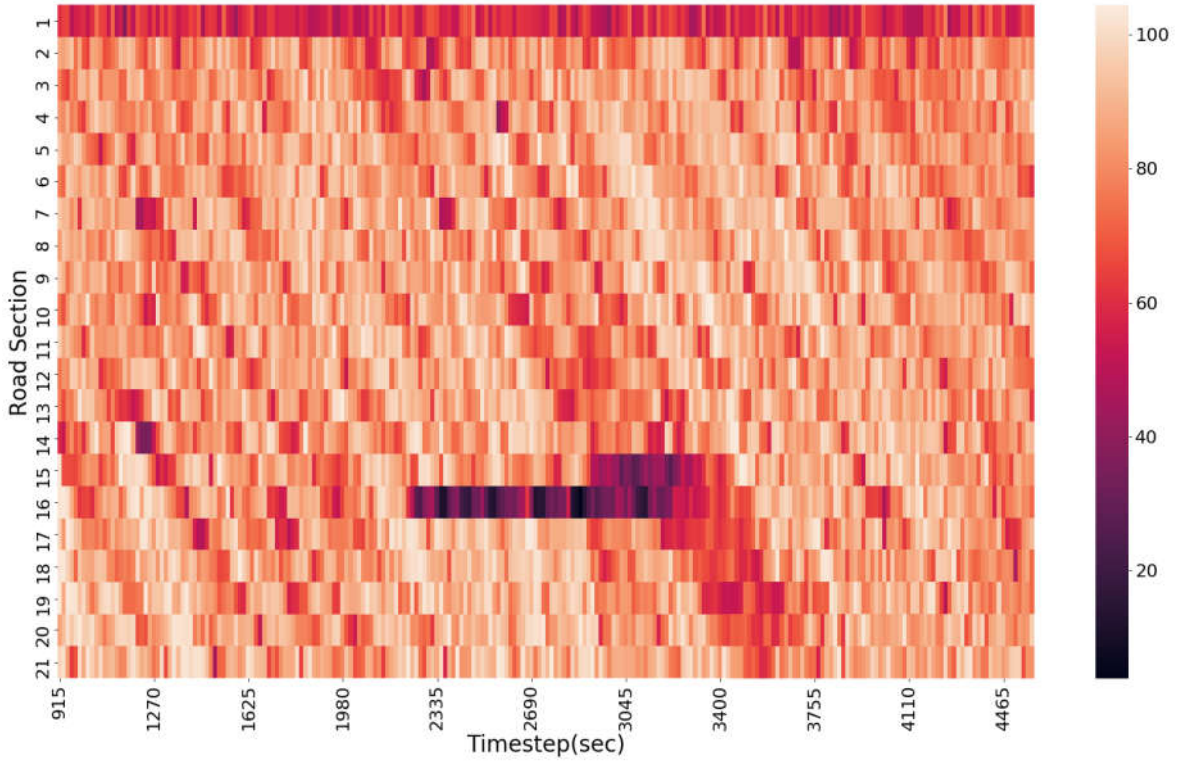

Figure C.6. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 50%.

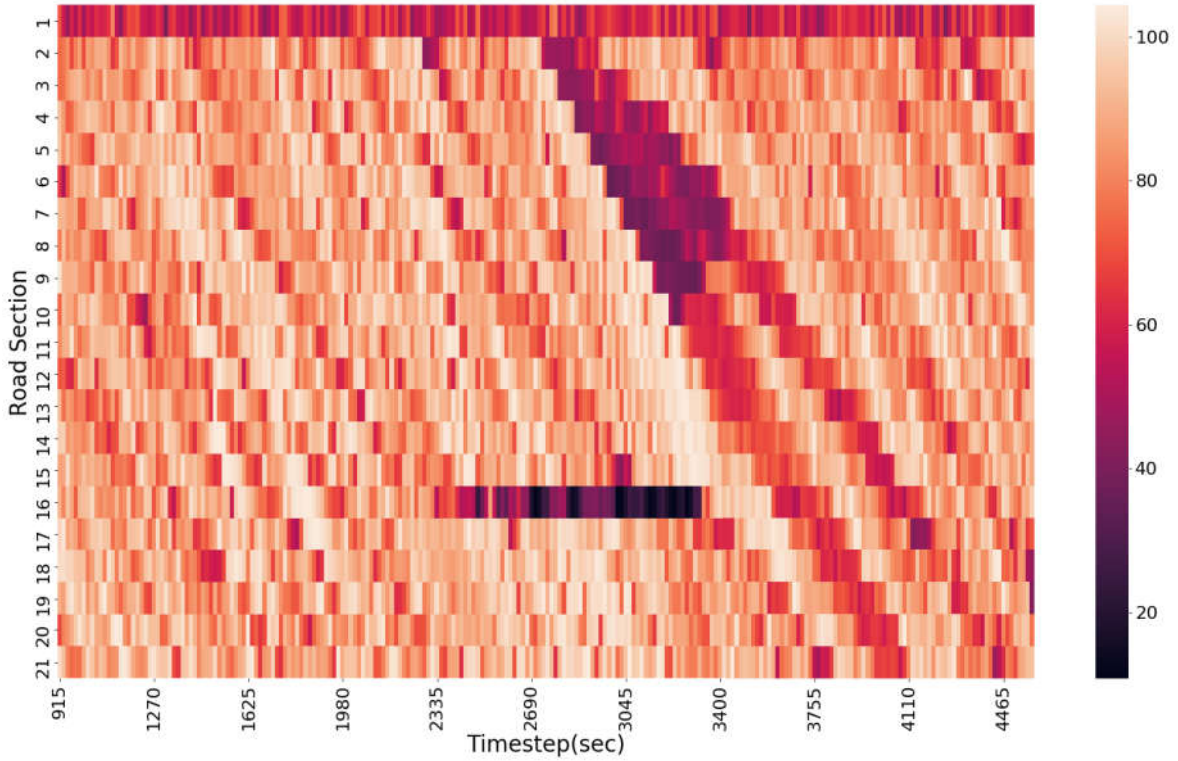

Figure C.7. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 60%.

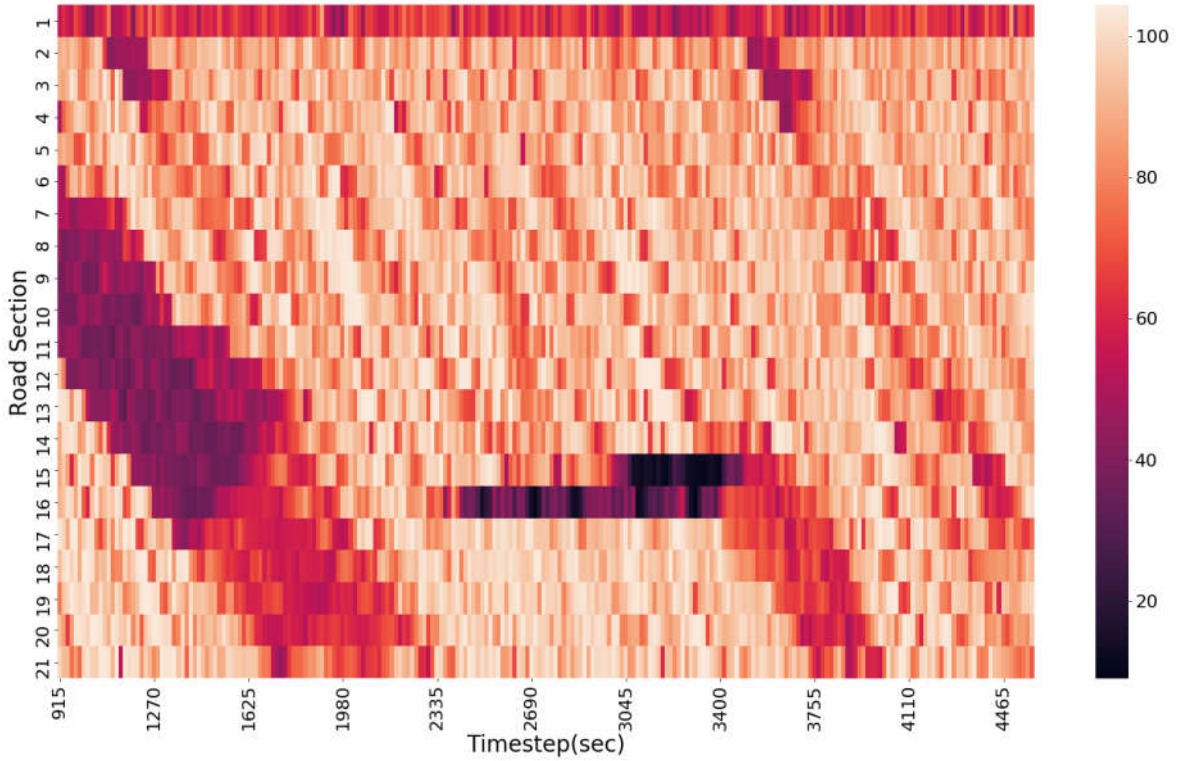

Figure C.8. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 70%.

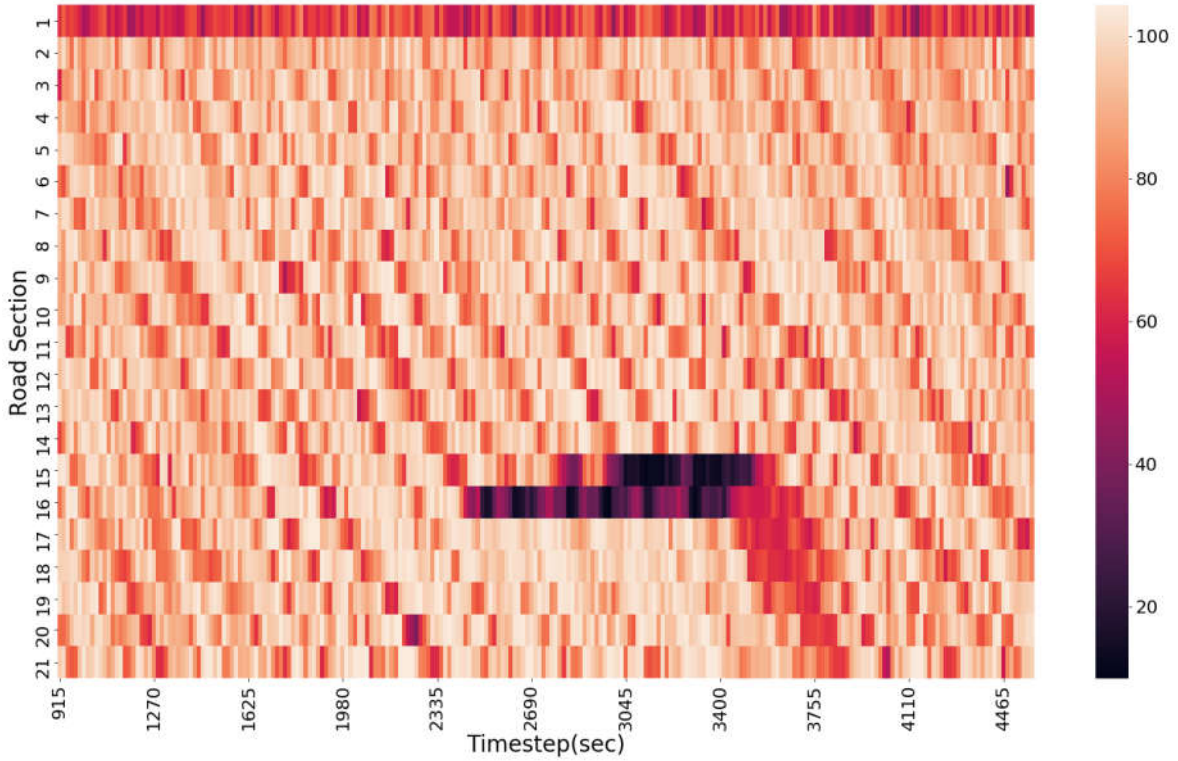

Figure C.9. Avg. Speed Heatmap When Accident in Lane 1 & Connected Autonomous Vehicle Ratio Is With 80%.

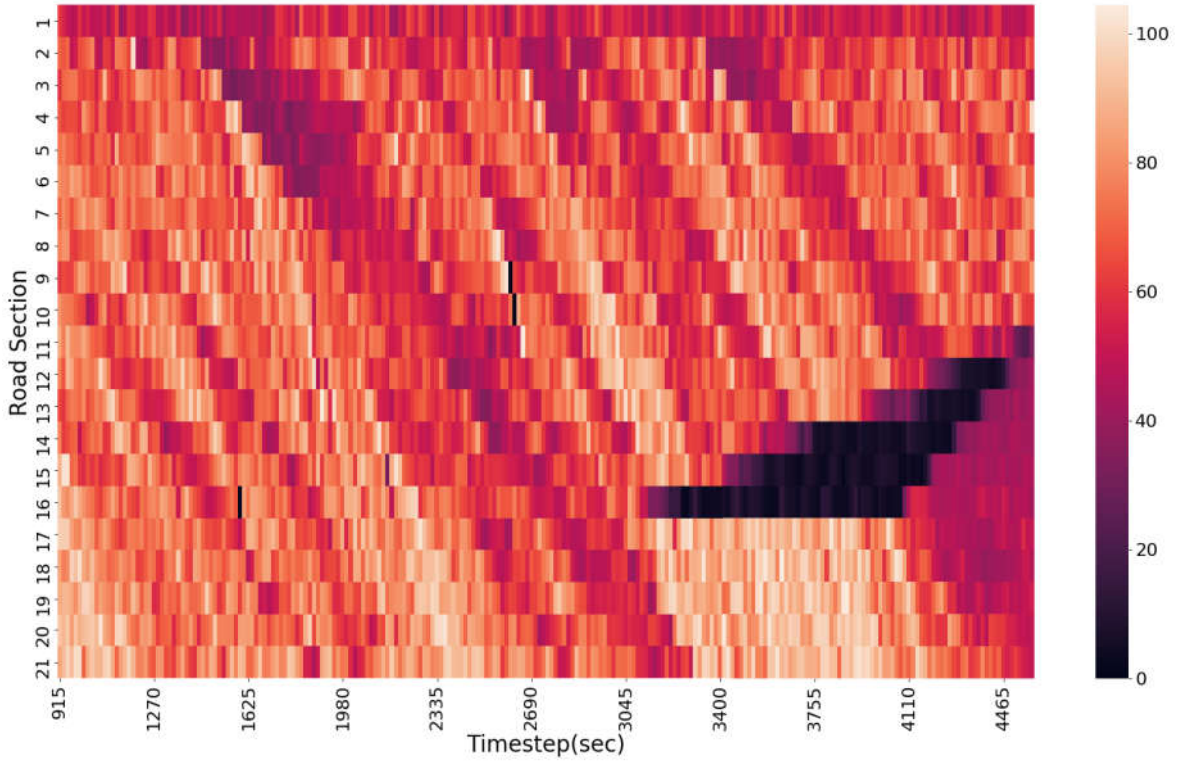

Figure C.10. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 0%.

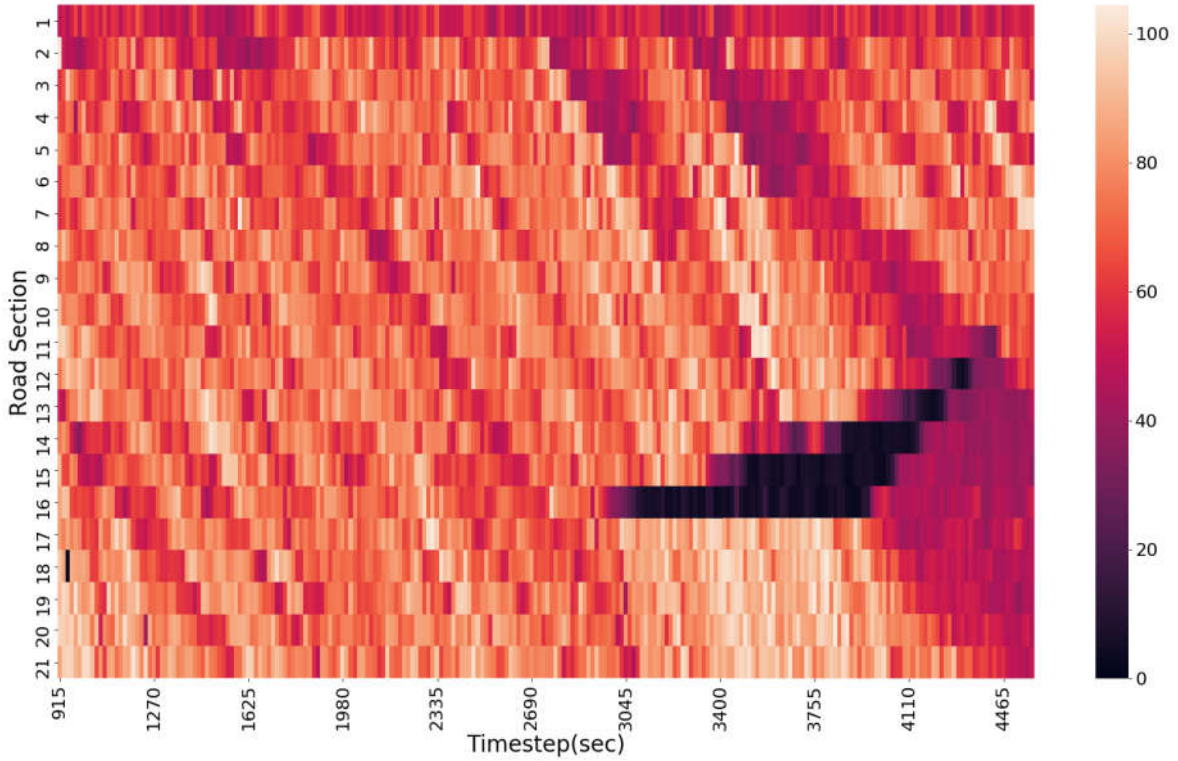

Figure C.11. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 10%.

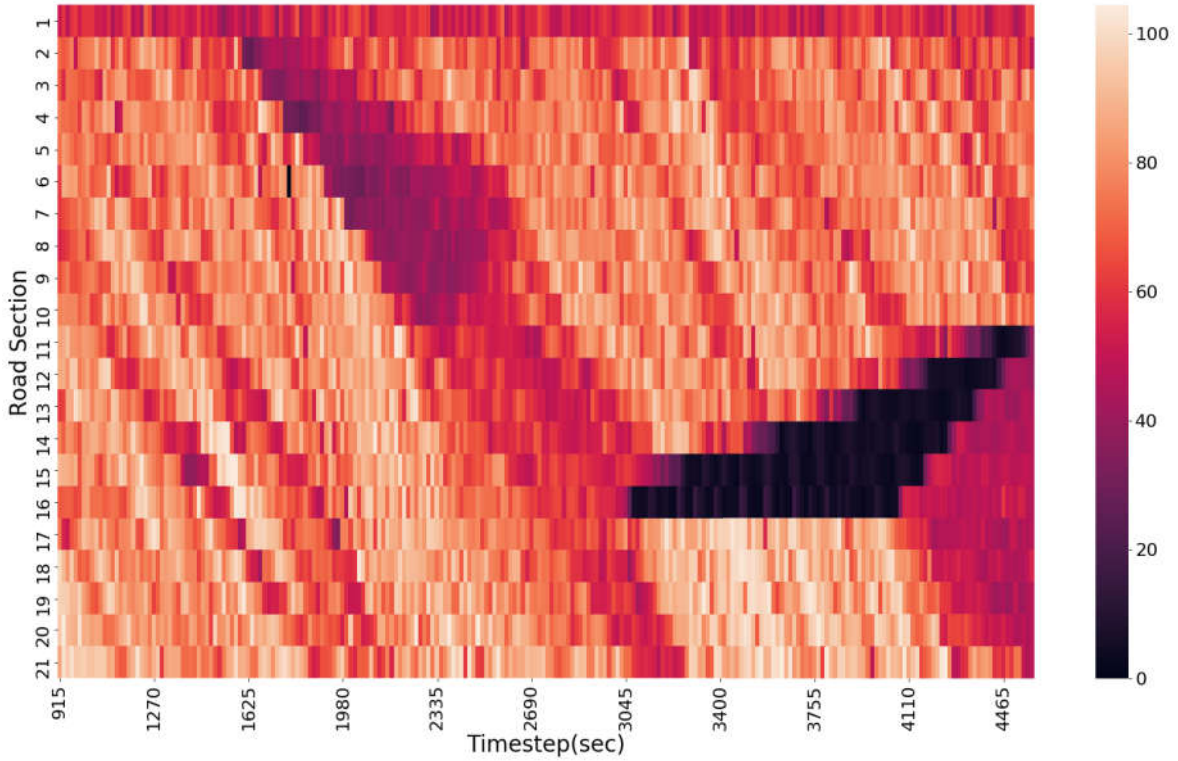

Figure C.12. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 20%.

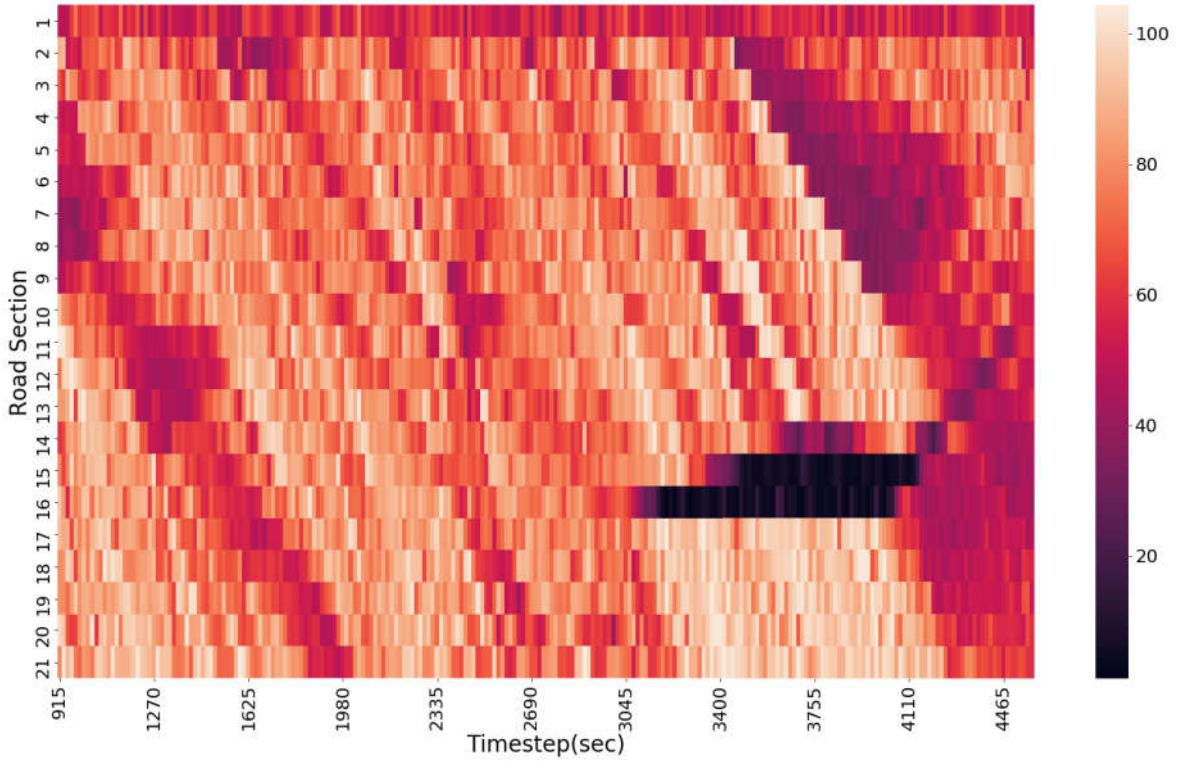

Figure C.13. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 30%.

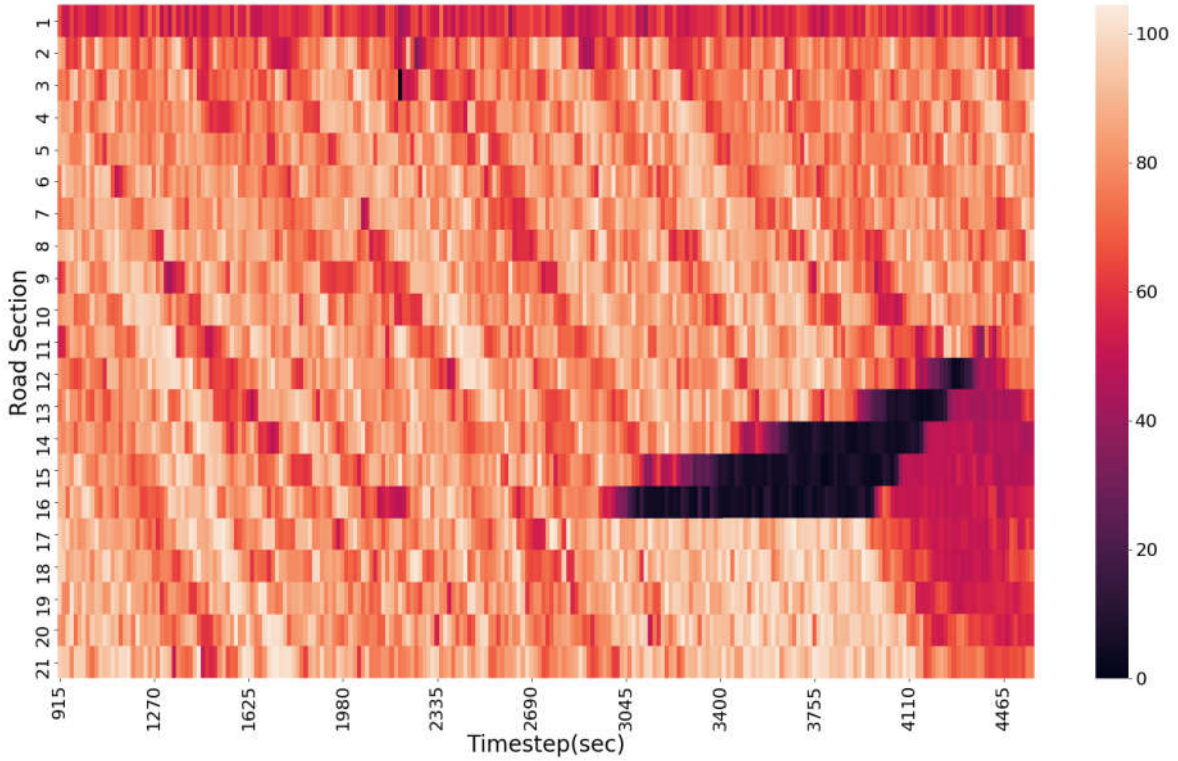

Figure C.14. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 40%.

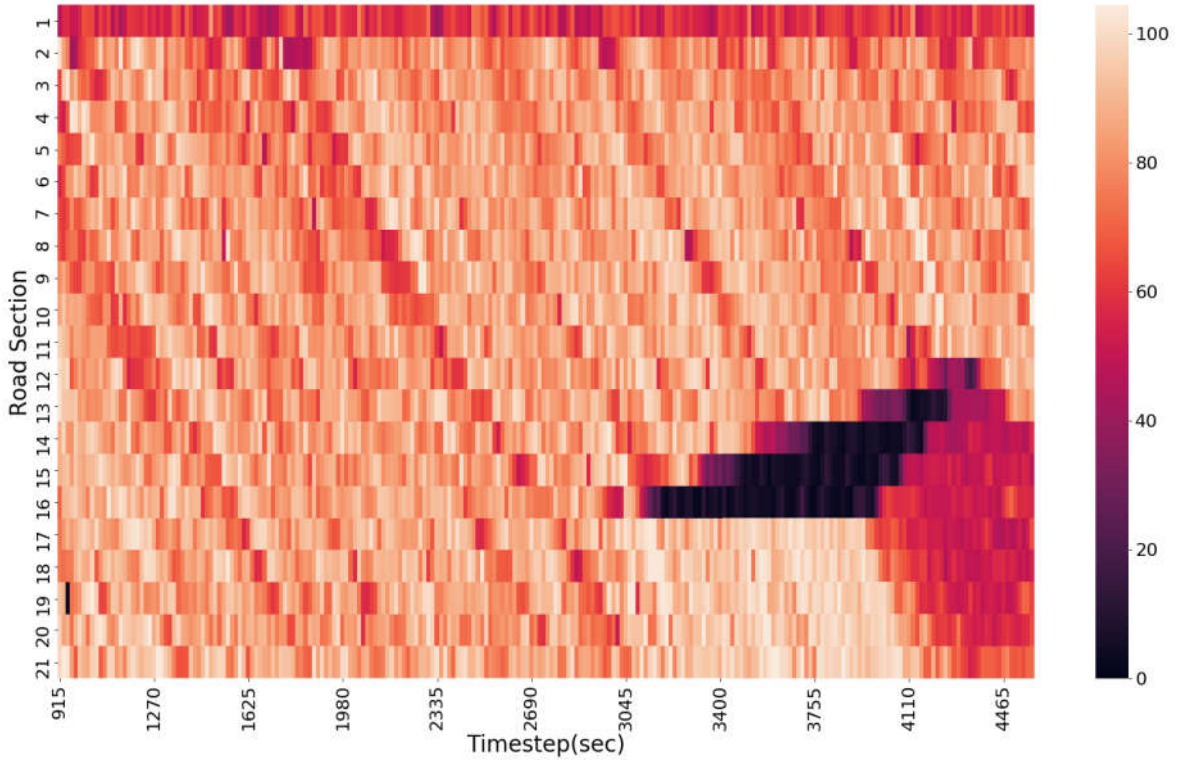

Figure C.15. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 50%.

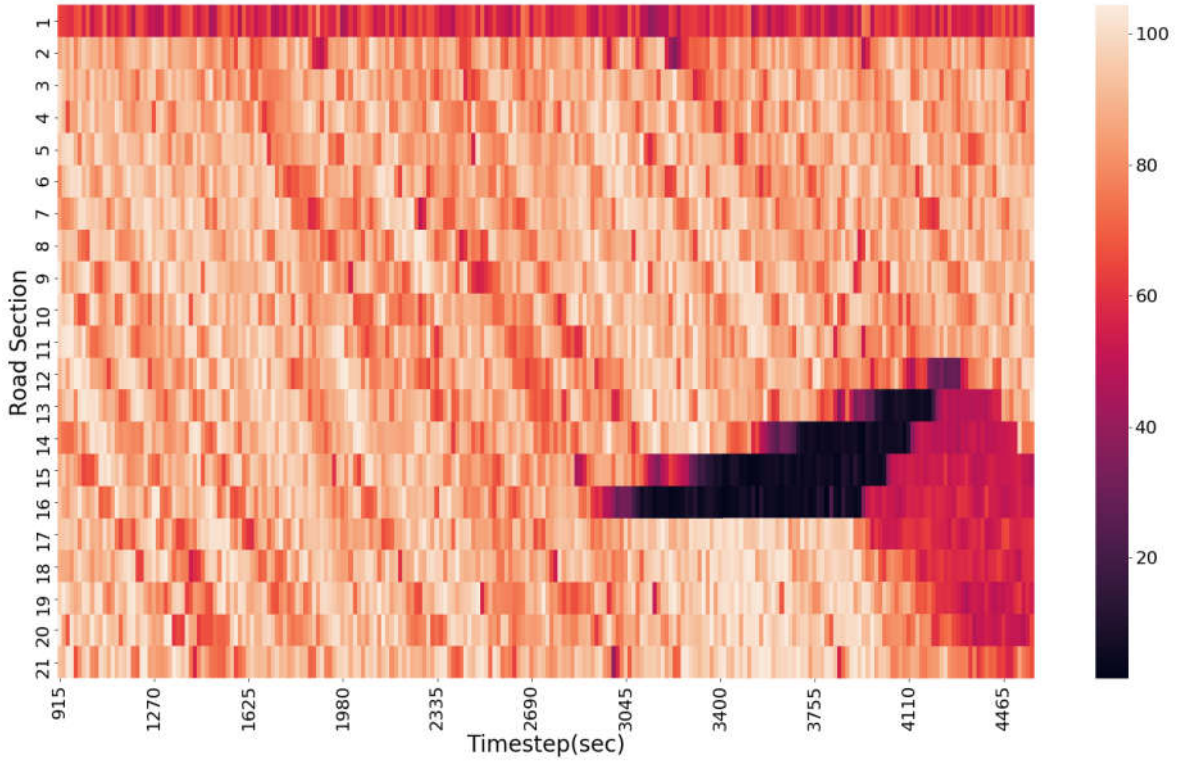

Figure C.16. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 60%.

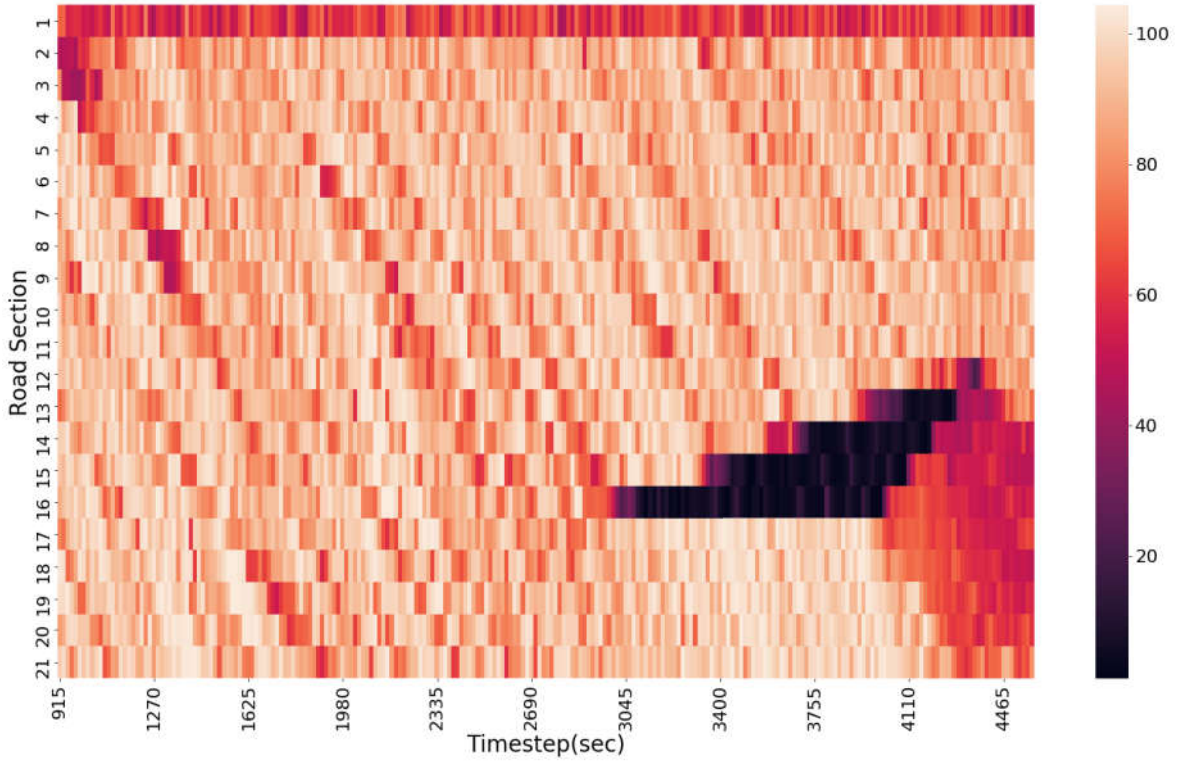

Figure C.17. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 70%.

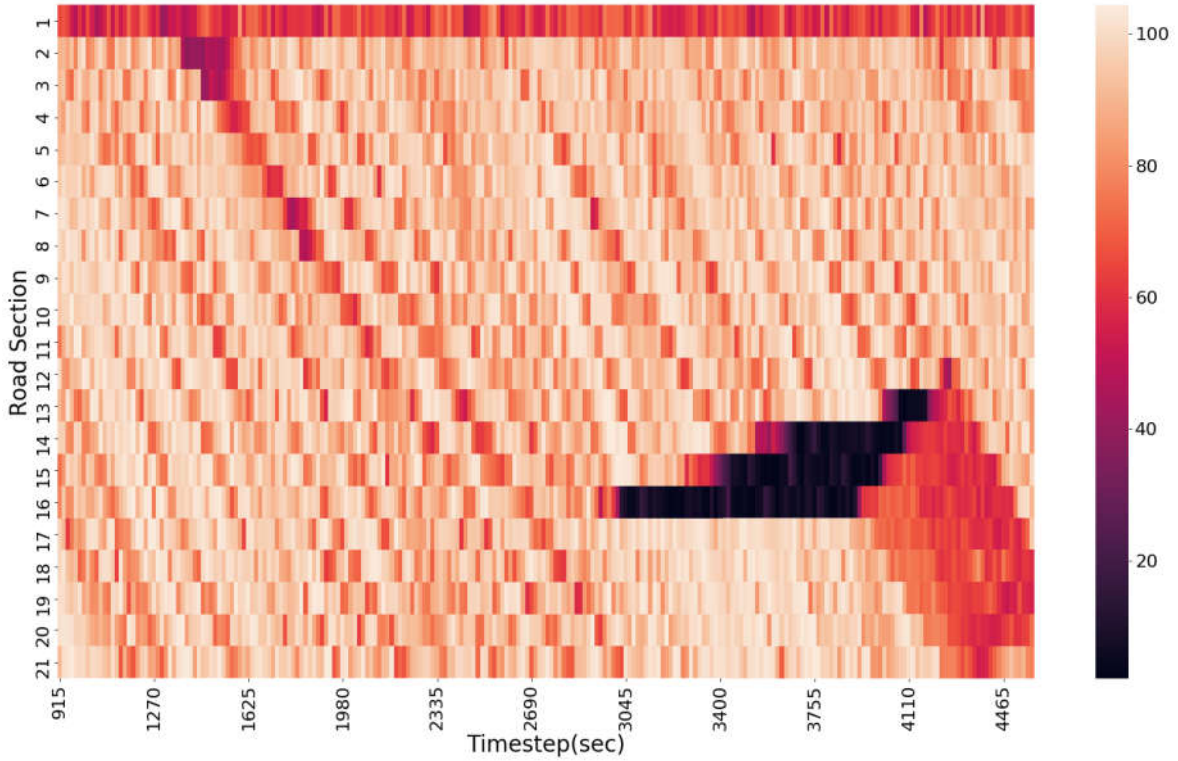

Figure C.18. Avg. Speed Heatmap When Accident in Lane 2 & Connected Autonomous Vehicle Ratio Is With 80%.

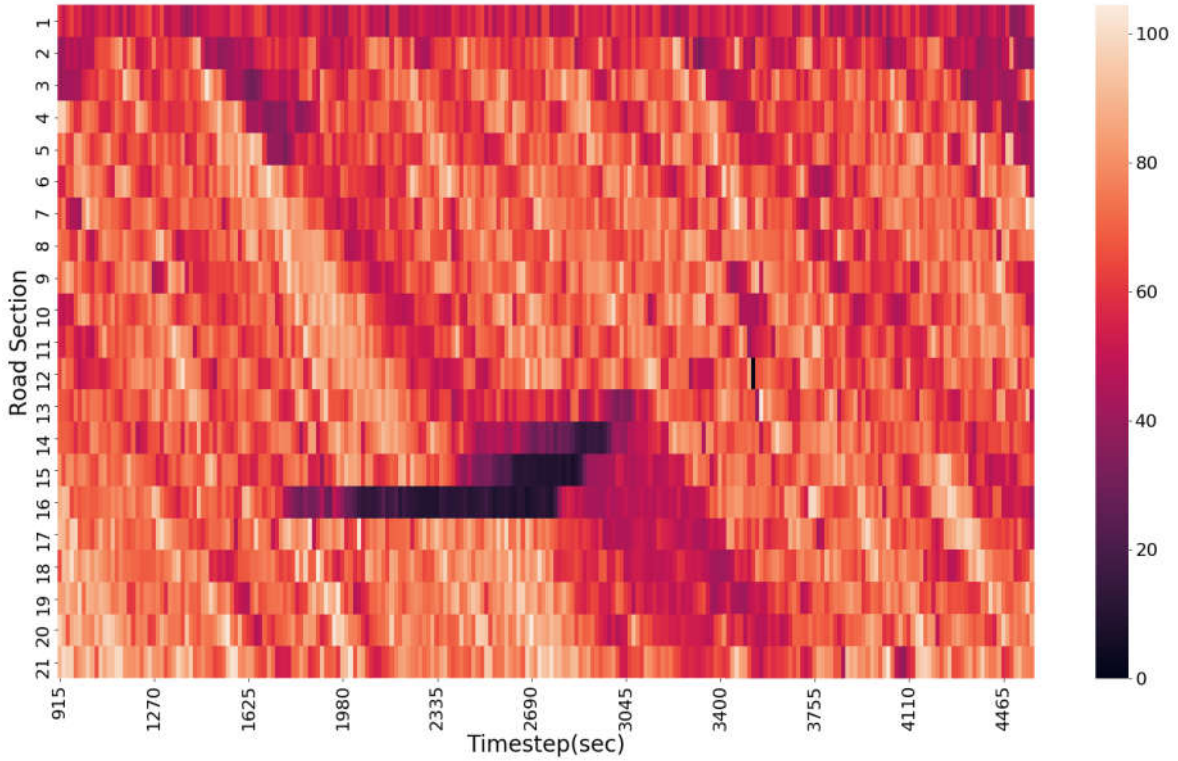

Figure C.19. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 0%.

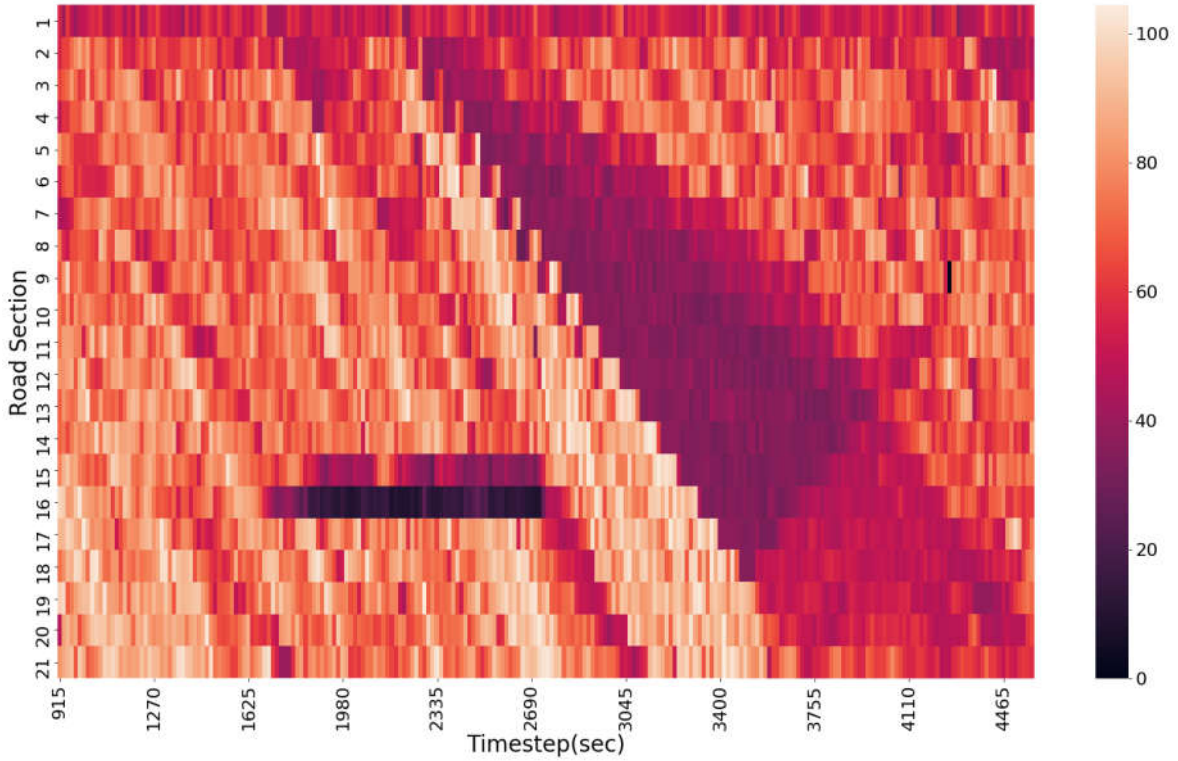

Figure C.20. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 10%.

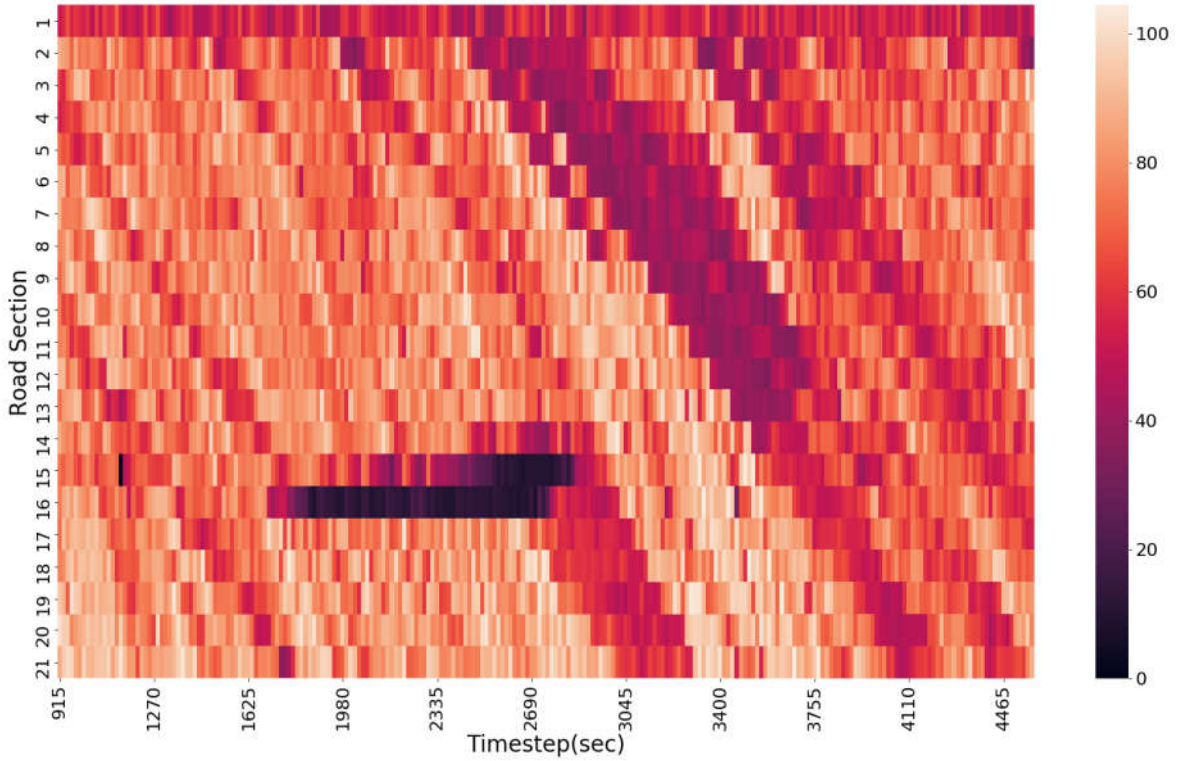

Figure C.21. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 20%.

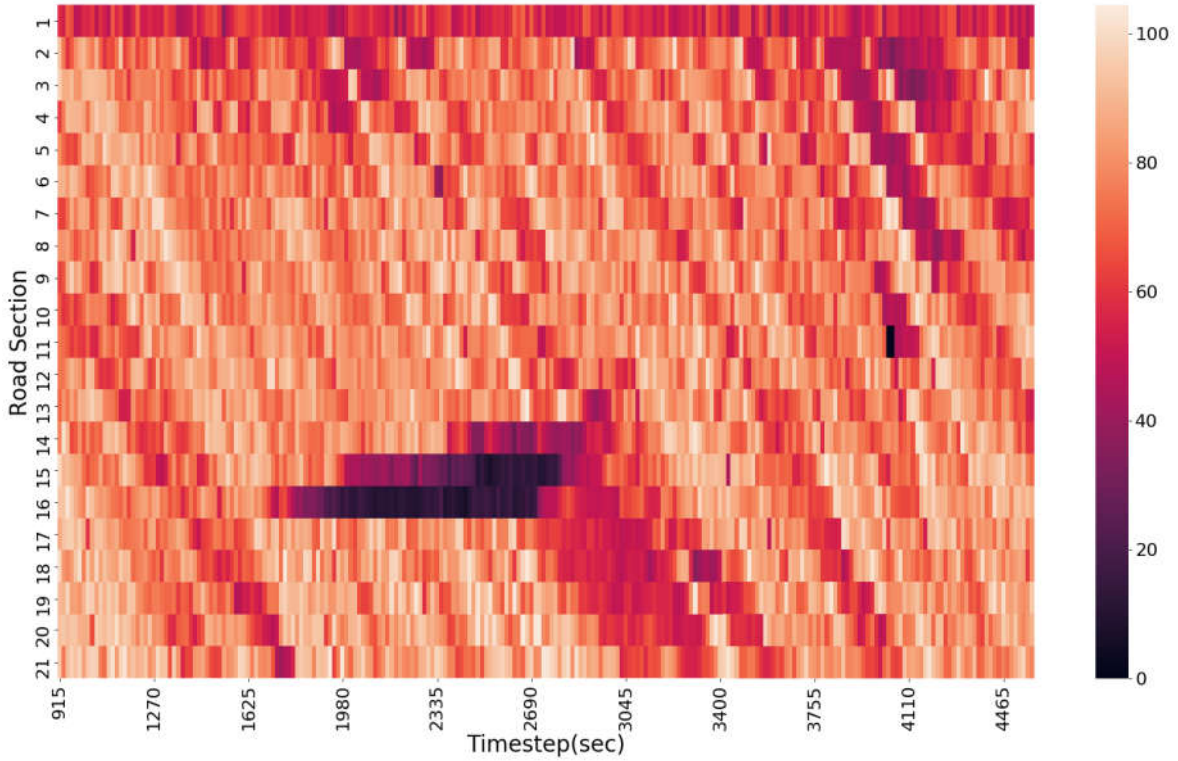

Figure C.22. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 30%.

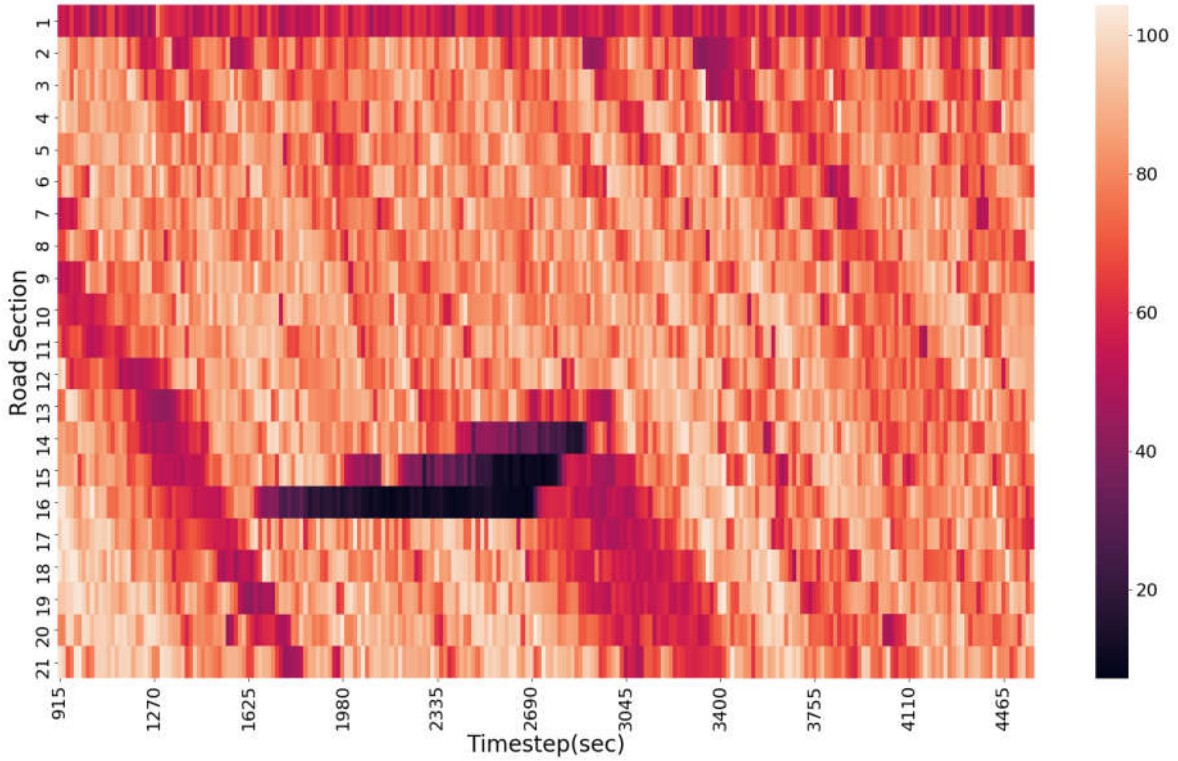

Figure C.23. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 40%.

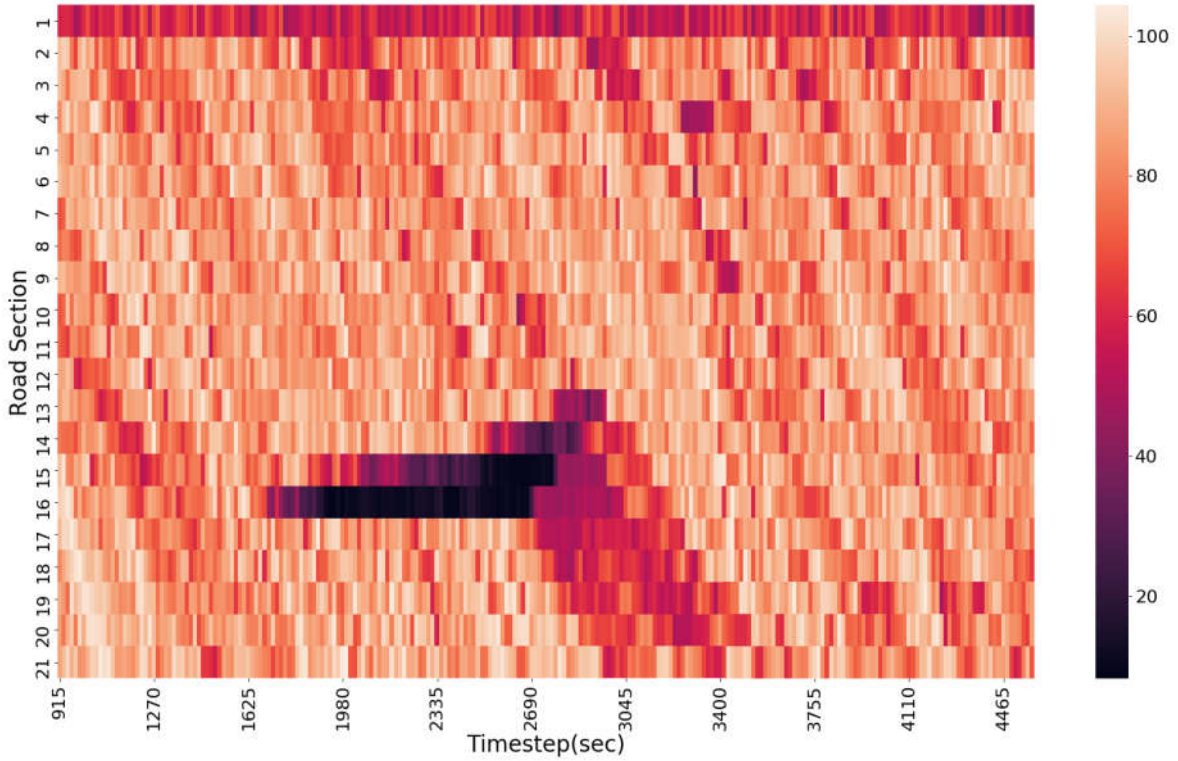

Figure C.24. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 50%.

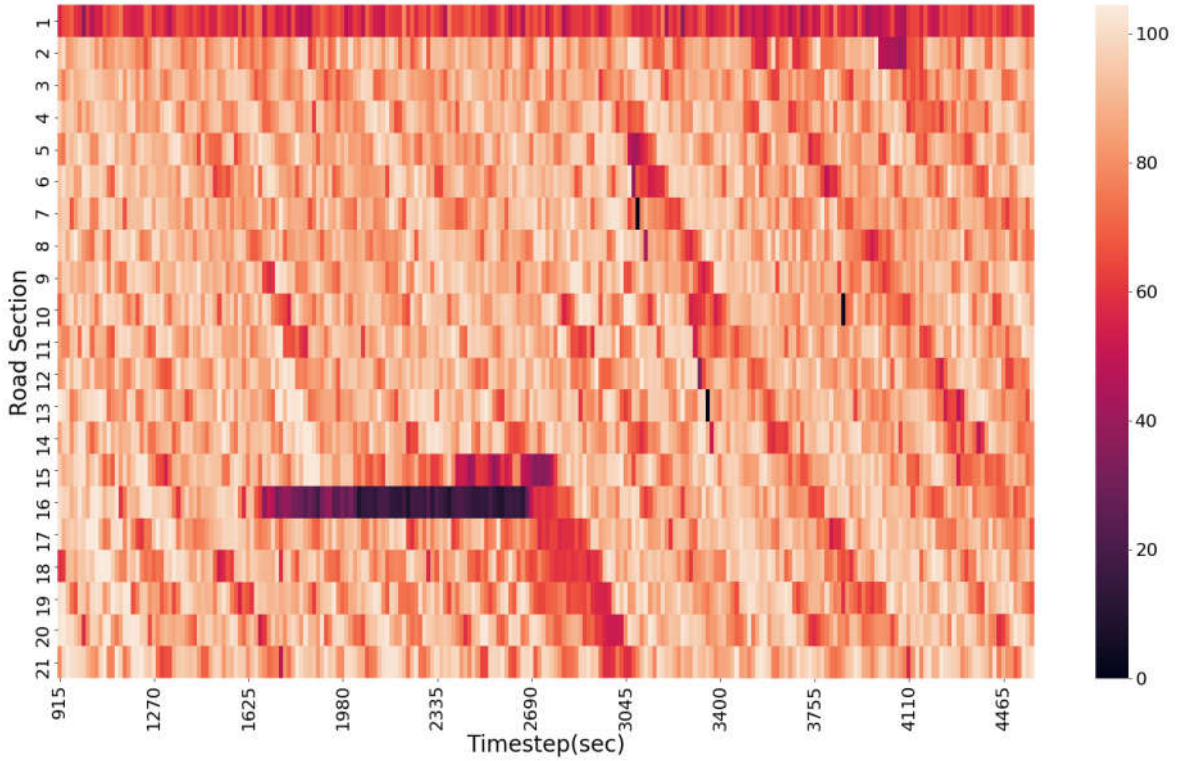

Figure C.25. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 60%.

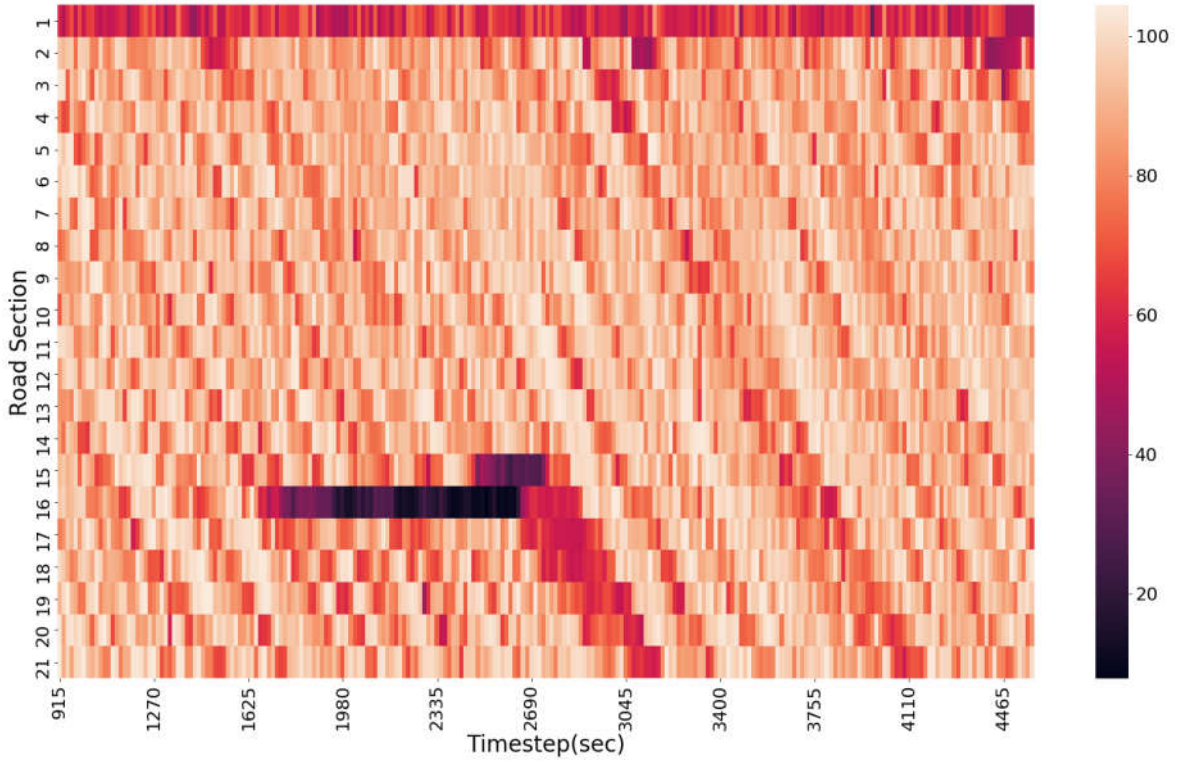

Figure C.26. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 70%.

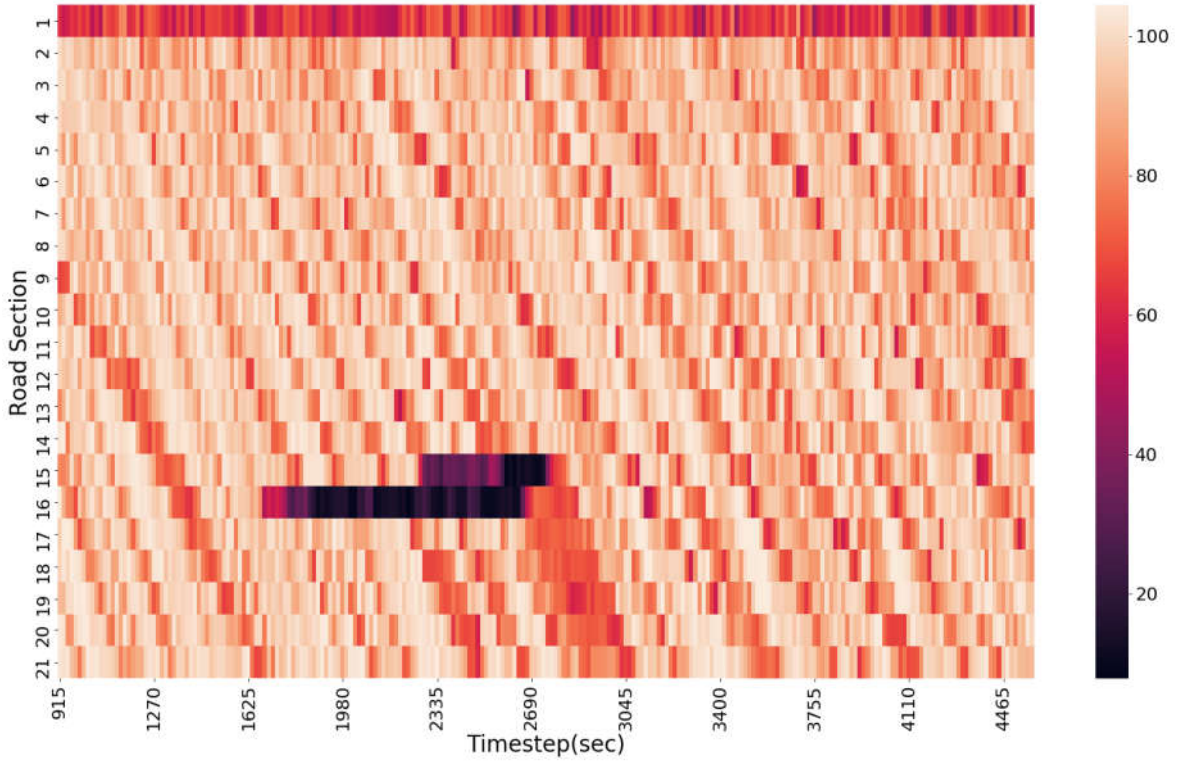

Figure C.27. Avg. Speed Heatmap When Accident in Lane 3 & Connected Autonomous Vehicle Ratio Is With 80%.