THE UNIVERSITY OF TEXAS AT AUSTIN
CENTER FOR TRANSPORTATION RESEARCH

TECHNICAL REPORT 0-7034-1
TxDOT PROJECT NUMBER 0-7034

An Exploration of the Use of Artificial Intelligence for EnhancedTraffic Management, Operations and Safety

Natalia Ruiz Juri, Ph.D.
Ken Perrine
Steve Boyles, Ph.D.
Kristie Chin, Ph.D.
Andrea Gold, Ph.D.
William Alexander
Gopindra Nair, Ph.D.
Jake Robbennolt
Morgan Avera, Ph.D.

January 20204

Published January 2024

https://library.ctr.utexas.edu/ctr-publications/0-7034-1.pdf

Technical Report Documentation Page

1. Report No. FHWA/TX-24/0-7034-1	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle An Exploration of the Use of Artificial Intelligence for Enhanced Traffic Management, Operations and Safety		5. Report Date Submitted: 1/24/2024 6. Performing Organization Code
7. Author(s) Natalia Ruiz Juri, Ph.D. https://orcid.org/0000-0001-7035-1867; Ken Perrine https://orcid.org/0000-0003-2272-0642; Steve Boyles, Ph.D.; Kristie Chin, Ph.D. http://orcid.org/0000-0002-8094-1008; Andrea Gold, Ph.D. https://orcid.org/0000-0003-4879-1182; William Alexander; Gopindra Nair, Ph.D.; Jake Robbennolt https://orcid.org/0000-0002-5106-2530; Morgan Avera, Ph.D. https://orcid.org/0000-0002-5210-9695.		8. Performing Organization Report No. 0-7034-1
9. Performing Organization Name and Address Center for Transportation Research The University of Texas at Austin 3925 W. Braker Lane, 4th Floor Austin, TX 78759		10. Work Unit No. (TRAIS) 11. Contract or Grant No. 0-7034
12. Sponsoring Agency Name and Address Texas Department of Transportation Research and Technology Implementation Division 125 E. 11th Street Austin, TX 78701		13. Type of Report and Period Covered Technical Report September 2019 – October 2023 14. Sponsoring Agency Code
15. Supplementary Notes Project performed in cooperation with the Texas Department of Transportation and the Federal Highway Administration.		

16. Abstract

This project explored the use and value of artificial intelligence and machine learning (ML) in transportation taking a multi-pronged approach that includes a literature review, a workshop, a survey, the development of 3 prototype ML models for four high-priority use cases, and the field testing of one of the prototyped models. Initial tasks were exploratory in nature and led to a better understanding of the current and prospective uses of ML in transportation, corresponding data needs, and specific use cases of interest to TxDOT.

Prototype model development was used to assess the value, challenges, and limitations of implementing several types of machine learning models to support the use cases prioritized by TxDOT. The prototypes leveraged emerging and traditional data sources: Wejo event data was used along CRIS data to build supervised and unsupervised learning models for understanding safety hot-spots and evaluating the effects of the pandemic on safety and traffic patterns; a microsimulation environment was used to explore the feasibility of adjusting traffic signal timing plans in real-time in a frontage road setting using reinforcement learning models; probe-based speeds from INRIX were combined with traffic volume data from TxDOT's ITS to generate short term travel time predictions on I-35, which could lead to more accurate driver information.

Short -term travel time prediction models were selected for field testing given the promising results found during the prototyping and the maturity and widespread availability of the involved data sources. In our preliminary models ML experienced travel time predictions were 40% more accurate during peak periods that those from traditional approaches.

Field testing included the training of additional models in Austin and El Paso, and the development of a framework to expedite model training, testing, and evaluation, and to support real-time deployment. Real-time predictions were shared with TxDOT through a web-based application that also facilitated model evaluation.

The performance evaluation of the newly trained models considered the ability to correctly identify the fastest route among competing alternatives, and predictive ML models were found to be correct more often than traditional approaches.

17. Key Words	18. Distribution Statement
Artificial Intelligence, Emerging Data, Safety, Innovation, Travel Time Information, Assessment of Pandemic Impacts, Connected Vehicle Data, ITS Data, Data Management, Format and Content Type.	No restrictions. This document is available to the public through the National Technical Information Service, Alexandria, Virginia 22312; www.ntis.gov.

19. Security Classif. (of report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	267	

THE UNIVERSITY OF TEXAS AT AUSTIN
CENTER FOR TRANSPORTATION RESEARCH

An Exploration of the Use of Artificial Intelligence for Enhanced Traffic Management, Operations and Safety

Natalia Ruiz Juri, Ph.D.
Ken Perrine M.S.
Jake Robbennolt
William Alexander
Steve Boyles, Ph.D.
Kristie Chin, Ph.D.
Andrea Gold, Ph.D.
Gopindra Nair, Ph.D.
Morgan Avera, Ph.D.

CTR Technical Report:	0-7034-1
Report Date:	Submitted: January 2024
Project:	0-7034
Project Title:	Exploring the Use of Artificial Intelligence to Leverage TxDOT Data for Enhanced Corridor Management and Operations
Sponsoring Agency:	Texas Department of Transportation
Performing Agency:	Center for Transportation Research at The University of Texas at Austin

Project performed in cooperation with the Texas Department of Transportation and the Federal Highway Administration.

Center for Transportation Research
The University of Texas at Austin
3925 W. Braker Lane, 4th floor
Austin, TX 78759

http://ctr.utexas.edu/

Disclaimers

Author's Disclaimer: The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official view or policies of the Federal Highway Administration or the Texas Department of Transportation (TxDOT). This report does not constitute a standard, specification, or regulation.

Patent Disclaimer: There was no invention or discovery conceived or first actually reduced to practice in the course of or under this contract, including any art, method, process, machine manufacture, design or composition of matter, or any new useful improvement thereof, or any variety of plant, which is or may be patentable under the patent laws of the United States of America or any foreign country.

Engineering Disclaimer

NOT INTENDED FOR CONSTRUCTION, BIDDING, OR PERMIT PURPOSES.
Research Supervisor: Natalia Ruiz Juri

Acknowledgments

The authors express appreciation to the TxDOT Project Director, Joanne Steel, and all of the Project Monitoring Committee Members including Jianming Ma, James Kuhr, David Freidenfeld, David Tidwell, Eduardo Perales, Farhan Khan, Jenny Li, Kevin Plumlee and Salvador Perez. We would also like to express our gratitude to the Austin and El Paso Districts for their support in accessing real time and archived data and to Brent Eastman at SWRI for his assistance in our use for the c2c data feed. The authors greatly appreciate for the support received from INRIX for the programmatic use of their data, as well as the support received from Wejo in the access to and interpretation of their sample data.

Table of Contents

Chapter 1. Introduction 1
Chapter 2. Literature Review 4
2.1. The State of Practice in Artificial Intelligence 4
2.1.1. Machine Learning 6
2.1.2. Deep Learning 8
2.1.3. Evolving AI 9
2.2. Transportation Data Sources 9
2.2.1. Passive Sensors 10
2.2.2. Semi-Passive Sensors 11
2.2.3. Active Sensors 12
2.2.4. Non-Sensor Data (Contextual Data) 12
2.3. Strategic Initiatives 13
2.4. Applications of AI in Transportation 14
2.4.1. System and Service Planning 14
2.4.2. Asset Management 16
2.4.3. System Operations 17
2.4.4. Communications & Information 21
2.4.5. Business Administration 23
2.4.6. Public Safety and Enforcement 25
2.5. Methods and Tools 27
2.5.1. Libraries, Platforms, and Tools 27
2.5.2. Databases & Data Access 29
2.5.3. Computer Resources and Environments 31
2.6. Conclusions 32
Chapter 3. Data Survey 33
3.1. Overview of Data Access Points and Data Streams 33
3.2. Lonestar 34
3.3. TxDOT Data Lake 35
3.4. Highway Condition Reporting System 36
3.5. Statewide Traffic Analysis and Reporting System (STARS) 37
3.6. Crash-Records Information System (C.R.I.S.) 38

3.7. TxDOT Open Data Portal 39
3.8. Third-Party Traffic Speeds and Travel Times 40
3.8.1. Inrix 40
3.8.2. Waze for Cities Data 41
3.9. Transit Data 43
3.10. Experimental/Project-Specific Data with Limited Availability 44
3.11. Conclusions 44
Chapter 4. Use Case Prioritization 46
Chapter 5. A Strategic Approach to Data Acquisition 49
5.1. Survey Development 50
5.1.1. Design 50
5.1.2. Refinement 51
5.1.3. Distribution 51
5.2. Survey Results 52
5.2.1. Background Data 52
5.2.2. Priority Applications 53
5.2.3. Respondent Sentiments 55
5.3. Data Acquisition Strategies 59
5.3.1. Digital Infrastructure and Data Assets 59
5.3.2. Personnel 60
5.3.3. Policy and Planning 61
5.4. Conclusion 62
Chapter 6. Data Collection for Prototypes 64
6.1. Data from Smart Work-Zone Trailers 64
6.1.1. Data Sampling and Processing 66
6.2. Probe-based Speed Data 66
6.2.1. Data Sampling and Processing 67
6.3. Data from Permanent ITS Sensors 68
6.3.1. Data Sampling and Processing 68
6.4. High Frequency Connected Vehicle Data | Basic Safety Message 69
6.4.1. Data Sampling and Processing 69
6.5. Additional Data on Austin's Arterial System 70
6.5.1. Data Sampling and Processing 70

Chapter 7. Safety Analysis Using Unsupervised Learning 71
7.1. Background 71
7.2. Research Question 72
7.2.1. Geography 72
7.2.2. Time Period 72
7.3. Data Collection & Management 72
7.3.1. Wejo Data 72
7.3.2. Roadway Geometry Data 75
7.3.3. CRIS Data 76
7.4. Analysis 77
7.4.1. Spatial Analysis 77
7.4.2. Cluster Analysis 77
7.5. Results & Key Findings 80
7.5.1. Comparison of Before and After Stay-at-Home Order 80
7.5.2. Cluster Analysis Results 84
7.6. Conclusions 93
7.7. Opportunities & Recommendations 94
Chapter 8. Travel Patterns Using Supervised Learning 96
8.1. Data 96
8.1.1. Wejo CV Data 96
8.2. Analysis Methodology 100
8.2.1. Measure of Fit 101
8.3. Dataset Preprocessing 101
8.3.1. Removing Outliers 101
8.3.2. Removing Biased Records 101
8.3.3. Removing Block-Groups with Low Trip Counts 102
8.4. Training and Testing Method 102
8.5. Modeling Frameworks 103
8.5.1. Linear Regression 103
8.5.2. Decision Tree Regression 104
8.5.3. Random Forest Regression 105
8.5.4. Multilayer Perceptron Regression 105
8.6. Model Results and Discussion 106

8.7. Conclusions 109
Chapter 9. Traffic Signal Timing Using Reinforcement Learning..... 110
9.1. Components of Reinforcement Learning..... 110
9.1.1. Environment..... 110
9.1.2. Performance Element..... 111
9.1.3. Learning Mechanism 111
9.1.4. Problem Generation..... 111
9.1.5. Evaluation Standard 112
9.2. Initial RL Model 112
9.2.1. Scenario Development 112
9.2.2. Training Environment Development 113
9.2.3. Results and Discussion 114
9.3. Further RL Modeling..... 114
9.3.1. State evaluation 116
9.3.2. Action choice..... 116
9.3.3. Performance..... 118
9.3.4. Extended testing 120
9.4. Conclusions 125
9.5. Pathway to Implementation..... 126
9.5.1. Site Location..... 127
9.5.2. Hardware..... 128
9.5.3. Software..... 129
9.5.4. Training 131
9.5.5. Experiment Monitoring 132
9.5.6. Evaluation..... 134
9.5.7. Resiliency..... 135
9.5.8. Conclusion..... 135
Chapter 10. Short-term Travel-time Prediction..... 136
10.1. Preliminary Model Development 137
10.1.1. Experiment design..... 138
10.1.2. Data..... 139
10.1.3. Model development and training..... 144
10.1.4. Model evaluation 147

10.1.5. Discussion 158
10.2. Development of Models for Field Testing 159
10.2.1. Experimental Design 159
10.2.2. Model Development and Training 163
10.2.3. Data 166
10.2.4. Phase I Models 168
10.2.5. Phase II Models 174
10.3. Continuous Evaluation Framework and Results 179
10.3.1. Performance Evaluation Approach 179
10.3.2. Lookback Windows 181
10.3.3. Metrics 181
10.3.4. Sample of Performance Metric Computation 184
10.3.5. Real Time Evaluation of Phase I Models 186
10.3.6. Real-time Evaluation of Phase II models 194
10.3.7. Predicting Travel Time Differences Across Competing Routes 197
10.4. Conclusions 213
10.5. Pathway to Implementation 214
10.5.1. Model Training 215
10.5.2. Collection of historical data 216
10.5.3. Automation and refinement of model training 217
10.5.4. Data Pipelines 217
10.5.5. Model Deployment 218
10.5.6. Model Evaluation 218
10.5.7. Summary and Discussion of Required Computational Resources . 219
Chapter 11. Conclusions 221
References 226
Appendix A. Survey Results 235
Appendix B. Dynamic Travel Times 242
Appendix C. Implementation Results of Phase II Models 243
Appendix D. Incorrect Shortest Route Predictions (Atypical) 270
Appendix E. Real-Time Model Performance 272

List of Tables

Table 1. Data Elements from Passive Sensors 11
Table 2. Data Elements from Semi-Passive Sensors 12
Table 3. Data Elements from Active Sensors 12
Table 4. Example Data Elements from Non-Sensor Data 13
Table 5. Data considered in Data Survey. 34
Table 6. Lonestar XML data types. 35
Table 7. SWZT Data Tables 64
Table 8. INRIX Data Tables 68
Table 9. Crash Severity Codes 76
Table 10. Top Six Contributing Factors 76
Table 11: Sample Table of Data 79
Table 12: March Results Summary 85
Table 13: March Results 85
Table 14: April Results Summary 87
Table 15: April Results 89
Table 16: Coefficients Estimated Using Linear Regression and the Percentage Change in Coefficients between March and April 106
Table 17: Example of static and dynamic travel time computation 138
Table 18: Testing metrics for the trained models 184
Table 19: Metrics for the Southbound SH 130 model 185
Table 20: Metrics for the Southbound SH 130 naïve prediction method 185
Table 21: Route testing summary. 188
Table 22: Model testing summary 189
Table 23: Percentage of time incorrect predictions of shortest route are made using naïve method. 206
Table 24: Percentage of time incorrect predictions 30-minutes in the future of shortest route are made using naïve method. 206
Table 25: Comparison of percentage of time incorrect predictions of shortest route are made using naive method versus ML models. 212
Table 26: Comparison of percentage of time incorrect predictions of shortest route are made using naive method versus ML models (atypically high travel times). 270

Table 27: Comparison of percentage of time incorrect predictions of shortest route are made using naive method versus ML models (atypically low travel times). 271

Table 28: MAE for travel time predictions models run in real time..... 272

List of Figures

Figure 1. Relationship of AI, ML, and DL (Argility, n.d.). 5
Figure 2. Comparison of NoSQL & Relational Databases (Bispo and Andres). 30
Figure 3. Drive Texas User Interface (http://drivetexas.org/). 36
Figure 4. STARS II user interface. 37
Figure 5. User interface for online C.R.I.S. data query tool. 39
Figure 6. Landing page of TxDOT's Open Data Portal. 40
Figure 7. Data shared through the Waze data feed. 42
Figure 8. Example of Waze for Cities Data Dashboard. 42
Figure 9. Transit data standards. 43
Figure 10. Experience and knowledge levels among respondents. 52
Figure 11: Priority application choices. 54
Figure 12: Relative priority rankings of top applications (darker shade indicates a higher ranking). 55
Figure 13: Factors that influenced top application selection and ranking. 56
Figure 14: Results for barriers to implementation. 57
Figure 15: Factors when evaluating new data source or analysis tool. 58
Figure 16. Location of SWZTs and selected permanent ITS sensors in the Austin District. 65
Figure 17. Example of availability of probe-based speed data along I-35 when accessed through the NPMRDS (a) and when purchased from the provider (b). 67
Figure 18. Movement Data Diagram. 73
Figure 19. Example of movement data point. 74
Figure.20: Example Event Data Point. 75
Figure 21: Heat Map of March Hard Acceleration Events. 81
Figure 22: Heat Map of April Hard Acceleration Events. 81
Figure 23: Heat Map of March Hard Braking Events. 82
Figure 24: Heat Map of April Hard Braking Events. 82
Figure 25: Heat Map of Speeding Event (L: March, R: April). 83
Figure 26: Heat Map of March Collisions. 83
Figure 27: Heat Map of April Collisions. 84
Figure.28: March Dendrogram. 86

Figure 29: April Dendrogram. 89
Figure 30: March Clusters (L: 1st to 15th St, R: Airport to US290). 90
Figure 31: April Clusters (L: 1st to 15th St, R: Airport to US 290). 91
Figure 32: Clusters from 1st St to 11th St (L: March, R: April). 92
Figure 33: Clusters at US 290 Interchange (L: March, R: April). 93
Figure 34 Trip Frequency (trips/h) Plotted at 15-minute Intervals for March and April 97
Figure 35. Locations of Trip Generation. 98
Figure 36. Census-Block-Group Level Trip Frequency (h-1). 99
Figure 37. Change in Trip Frequency per Block-Group – Absolute (Trips/mi^2/h) and Percentage. 99
Figure 38 Dataset Split into Training, Testing, and Validation Datasets. 103
Figure 39 Example Decision Tree. 104
Figure 40 Effect of Predictor Variables on an “Average” Block-Group. 108
Figure 41 R^2 for Estimated Models with Respect to Training and Testing Datasets. 109
Figure 42: Example Scenario Illustration. 113
Figure 43: PTV VISSIM Microsimulation Software. 115
Figure 44. Ring-barrier timing plan visualization (credit: Federal Highway Administration). 117
Figure 45. Econolite ASC/3-2100 virtual traffic signal controller 117
Figure 46: Reward distribution boxplots for four timing policies. 119
Figure 47. Comparison of selected cycle lengths during fifteen episodes before and after training. 120
Figure 48. VISSIM network model with freeway and "frontage road" in median intersecting a cross street. 121
Figure 49. Reward collected in simulations without a blockage. 123
Figure 50. Delay accumulated in simulations without a blockage. 123
Figure 51. Equity (Gini) coefficient for simulations without a blockage (lower is better). 124
Figure 52. Reward collected in simulations with a blockage. 124
Figure 53. Delay accumulated during scenarios with a blockage. 125
Figure 54. Equity (Gini) coefficient for simulations with a blockage (lower is better). 125
Figure 55. Field implementation pathway 127
Figure 56. INRIX segments and smart work zone trailer location. 139
Figure 57. Histogram of segment lengths. 140

Figure 58. Dynamic travel time patterns in 2019 using INRIX data. 141

Figure 59. Average 15-minute travel-time and corresponding standard deviation for 2 days of the week (2019). 142

Figure 60. Difference between average five-minute travel times in 2019 and corresponding mean values for 2019. Negative values reflect day/time combinations in which travel times are lower than expected; positive values represent times during which travel times exceeded typical values. 142

Figure 61. Distribution of 5-minute travel times based on their proximity to the corresponding average (expected) value in 2019. The "0" category corresponds to travel times that are within one standard deviation of the expected value and may be considered “typical” values. 143

Figure 62. General structure of recurrent neural networks. 145

Figure 63. RNN architecture. 146

Figure 64. Short-term segment-level forecast MSE comparison 148

Figure 65. Dynamic travel-time mean squared error and mean absolute error. 149

Figure 66. Dynamic travel-time mean squared error and mean absolute error during peak periods. 150

Figure 67. Prediction error by five-minute interval for all analyzed methods (northbound). 151

Figure 68. Travel times on January 29, 2019. 152

Figure 69. Travel times on January 9, 2019. 152

Figure 70. Travel times on December 25, 2019. 152

Figure 71. Distribution of prediction errors during "typical" traffic conditions. 153

Figure 72. Distribution of prediction errors during "atypical" traffic conditions. 154

Figure 73. Travel times on January 15, 2019. 154

Figure 74. Average prediction error by time of day under different typical (actual travel times within one standard deviation from the mean) and atypical (actual travel times between one and two standard deviations from the mean) traffic conditions on weekdays. 156

Figure 75. Prediction error by time of day for typical and atypical traffic conditions (northbound direction, weekdays). 157

Figure 76. Primary and secondary corridors in El Paso. 160

Figure 77. Primary and secondary corridors in Austin. 160

Figure 78. Location of existing DMS on I-10 and FM 375 through El Paso District. 161

Figure 79. Location of Existing DMS on I-35 and SH 130 in the Austin District. 161

Figure 80: Model training and execution workflows. 164

Figure 81: Share of INRIX data missing for each corridor. 167

Figure 82: R^2 values for models trained on a full year's data. 170
Figure 83: Mean squared error for models trained on a full year's data. 170
Figure 84: Mean absolute error for models trained on a full year's data. 171
Figure 85: Correlation plot of testing data for I-10 eastbound forecasts. 172
Figure 86: Testing R^2 values for models with (orange) and without (blue) C2C data. 173
Figure 87: Figure 10: Testing MSE values for models with (orange) and without (blue) C2C data. 173
Figure 88: Testing MAE values for models with (orange) and without (blue) C2C data. 174
Figure 89: Average MAE by route for predictions zero timesteps ahead (time t). 176
Figure 90: R^2 by route for predictions zero timesteps ahead (time t). 177
Figure 91: MAE (min) for timesteps t+0 to t+30 on I-35 SB. 178
Figure 92: R^2 for timesteps t+0 to t+30 on I-35 SB. 179
Figure 93: Path travel time calculation algorithm. 180
Figure 94: Simplified database schema. 181
Figure 95: Aggregate performance of selected ML model on I-10 EB. Real-time metrics computed between 11/17/2022 and 03/20/2023. Dots represent the number of data points in each dataset. 191
Figure 96: Performance by hour on I-10 EB for real-time models (11/17/2022-03/20/2023)... 192
Figure 97: Disaggregate performance (MAE) on I-10 EB 11/17/2022-03/20/2023. 193
Figure 98: MAE for naive and ML predictions using real-time data from January 2 - May 31, 2023. 196
Figure 99. Average Real Time Performance on SB I-35. 197
Figure 100: Predicted and realized travel times along the AUS NB corridor on April 1, 2023. .. 199
Figure 101: Comparison of travel-time differences between alternative routes computed using a naïve prediction method (current practice) and realized travel time (as a proxy for perfect travel time predictions). El Paso EB (negative means I-10 is faster, positive mean SL 375 is faster). .. 200
Figure 102: Comparison of travel-time differences between alternative routes computed using a naïve prediction method (current practice) and realized travel time (as a proxy for perfect travel-time predictions). El Paso WB (negative means I-10 is faster, positive means SL 375 is faster) 200
Figure 103: Comparison of travel-time differences between alternative routes computed using a naïve prediction method (current practice) and realized travel time (as a proxy for perfect travel time predictions). Austin NB (negative means I-35 is faster, positive means SH 130 is faster). 201
Figure 104: Mean and standard deviation of naïve prediction vs. realized travel-time differences between routes on each corridor 202

Figure 105: Mean absolute error in naive prediction of difference in travel time between routes. 203

Figure 106: Mean and standard deviation of naïve prediction vs. realized travel-time differences between routes on each corridor with a 30-minute delay (left) and MAE in future naive prediction of difference in travel time between routes at several time increments. 205

Figure 107: Mean of ML model prediction vs realized travel time differences between routes on each corridor with a zero-minute delay (left) and MAE in ML model prediction of difference in travel time between routes. 209

Figure 108: Mean of ML model prediction vs realized travel time differences between routes on each corridor with a 30-minute delay (left) and MAE in ML model prediction of difference in travel time between routes. 211

Figure 109. Implementation Tasks. 215

Figure 110. Data Workflows. 217

Figure 111. Example of Instantaneous and dynamic travel time computation. 242

Figure 112: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 SB. 243

Figure 113: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 NB. 244

Figure 114: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 SB. 245

Figure 115: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 NB. 246

Figure 116: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 EB. 247

Figure 117: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 WB. 248

Figure 118: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 EB. 249

Figure 119: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 WB. 250

Figure 120: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 SB. 251

Figure 121: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 NB. 252

Figure 122: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 SB. 253

Figure 123: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 NB. 254

Figure 124: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 EB. 255

Figure 125: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 WB. 256

Figure 126: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 EB. 257

Figure 127: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 WB. 258

Figure 128: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 SB. 259

Figure 129: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 NB. 260

Figure 130: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 SB. 260

Figure 131: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 NB. 261

Figure 132: MAE time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 EB. 262

Figure 133: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 WB. 262

Figure 134: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 EB. 263

Figure 135: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 WB. 264

Figure 136: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 SB. 264

Figure 137: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 NB. 265

Figure 138: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 SB. 266

Figure 139: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 NB. 266

Figure 140: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 EB. 267

Figure 141: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 WB. 268
Figure 142: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 EB. 268
Figure 143: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 WB. 269

Chapter 1. Introduction

Data analytics, automation, artificial intelligence (AI), and machine learning (ML) are changing transportation operations and service delivery. As the transportation enterprise becomes more data-driven and automated, there is the potential to realize greater safety and efficiency benefits for the traveling public (Merrefield, 2019). With the availability of richer, vast streams of data from Intelligent Transportation Systems (ITS), smart vehicles, connected infrastructure sensors, mobile phones, and more, the expected impacts on the ability to better manage roadways are considerable.

Given the volume and heterogeneity of currently available transportation data, traditional data analysis tools are often not sufficient to realize its potential. AI can drastically increase the ability of agencies to derive value from collected data. AI is a term that defines the design of computer systems or agents that can receive inputs from the environment and perform actions, behaving similarly to human agents (Rusell et al., 2016). From uncovering patterns in the system conditions that may lead to designing better operational strategies, to supporting enhanced decision making in complex situations, AI techniques may transform existing practices (FHWA, 2018). However, successfully selecting and implementing an AI approach for a specific use case requires careful consideration. Using out-of-the-box techniques without a proper understanding of the underlying assumptions and limitations may not lead to the expected results. Further, it is important to consider AI in the context of other modeling and analysis techniques available in the transportation industry, understand how they may complement each other, and assess when it is appropriate or beneficial to choose AI over other methods. The Performing Agency has conducted previous research that suggests that using a more complex model does not always guarantee better results (Boyles and Ruiz Juri, 2019).

AI techniques have been available since the 1950s, and applications in the transportation domain have been explored in the literature (Nguyen, Hoang, et al., 2018; Abduljabbar, Rusul, et al. 2019, Sadek, 2007, Zheng et al. 2008). However, advances in computing and communications along with an unprecedented availability of traffic data make the study of AI particularly relevant today.

The goal of this project is to summarize recent advances in AI as they are relevant to the receiving agency, and to demonstrate the potential benefits of these techniques through impactful use cases. Given the broad range of potential applications of AI, focus will be placed on those that may support Integrated Corridor Management (ICM). ICM is a suite of growing strategies to address roadway management and dependencies, and it is becoming more powerful as ITS data grows. The combination of ICM strategies with new data will present

opportunities to better manage traffic congestion and more effectively move people and goods through metropolitan networks and the system as a whole. Proactive ICM strategies enabled through AI may allow public agencies to use mechanisms such as managed lanes, alternate routing, and dynamic traffic control to achieve significantly greater levels of utilization of the existing roadway capacity, improve travel times, enhance safety, and increase reliability of travel. Some of the expected benefits from this project include:

Manage Data Better. Artificial intelligence is a complex and rapidly evolving field. Equipped with an understanding of basic concepts, data management systems, and advanced techniques, TxDOT can integrate artificial intelligence best practices into its Enterprise Information Management strategy.

Map AI Techniques to Transportation Challenges. By connecting transportation challenges directly to a suite of AI techniques, TxDOT will prioritize areas of high impact such as congestion relief or incident management for further development and investment.

Leverage Data to Gain Insights. TxDOT manages a significant amount of data and has the opportunity to apply artificial intelligence to predict commuter travel times, improve real-time decision-making for major events such as hurricane evacuations, or move freight more efficiently.

Improve Integrated Corridor Management. Many urban areas in Texas are facing gridlock, and TxDOT can apply artificial intelligence techniques to coordinate operational decisions across state and local networks and measure the quantifiable benefits for Texas travelers.

Learn and Scale. Starting with a basic application such as integrated corridor management offers TxDOT the ability to learn in a small-scale environment before expanding to a larger scope. As a result, TxDOT can maximize the value of its investment and decide how to guide future activities.

Data generated from Intelligent Transport Systems combined with advances in artificial intelligence will almost certainly advance ICM goals. Data complexity and volume will increase as ITS continues to develop, and that trend will make competence in the latest data analysis techniques even more crucial than it is today. Deep learning techniques are the state of the art when it comes to data analysis; they outperform all other methods at every machine learning task and AI task, and, on top of that, offer more capabilities. It will be essential to have experience with deep learning techniques in order to find patterns and features in data that will support more efficient and connected transportation systems, as other methods may not be able to keep up with the complexity of the data.

In order to realize the benefits of the latest in AI and machine learning research, it's important to understand the concepts that underlie AI tools, as well as the availability of alternative methods that AI may complement and/or replace. It is also critical to be aware of the characteristics, potential, and limitations of available data sources. The Performing Agency has

expertise in both the use of large and complex transportation data sets and the implementation of AI techniques. By learning from and optimizing an experience with one experimental AI ICM application, we can determine how to approach future applications, how many such applications we need, and what other infrastructure and technical improvements our system needs before broader application. As promising as deep learning techniques are, it's important to establish a system that can support complex AI applications end to end, so that they produce smart, actionable results to the right people.

With the maturation and deployment of ITS, opportunities will arise to investigate and prove the usefulness of artificial intelligence, machine, and deep learning for ICM and applications such as traffic-flow forecasting, traffic signal control, automatic vehicle detection, traffic incident processing and analysis, travel demand prediction, autonomous driving, route diversion and driver awareness, and driver behavior.

Value to TxDOT: With the significant volume of data being generated in transportation today and the advancements in data science, TxDOT has an opportunity to leverage the tools and techniques of artificial intelligence in order to remain at the cutting edge. In particular, there is an opportunity to apply artificial intelligence to complex transportation problems like congestion, traveler information, emergency response planning, and safety. These applications will enable TxDOT to find optimal solutions for making transportation systems and technologies more reliable and efficient.

The following chapters cover the activities conducted for this project, which include a literature review, a workshop, a data survey, the development of three prototype ML models for four high priority use cases, and the field testing of one of the prototyped models. The latter included the development of a framework to streamline data access, archiving, and results sharing. Initial tasks were exploratory in nature and led to a better understanding of the current and prospective uses of ML in transportation, corresponding data needs, and specific use cases of interest to TxDOT. Pathways to implementation are proposed for several of the prototyped models, and we also include the results of a TxDOT survey intended to provide insights into the development of a strategic data collection plan.

Chapter 2. Literature Review

Advances in computing and information technology are making possible more high-performance computing and applications in artificial intelligence (AI) which could offer many benefits to the transportation sector. These benefits include optimizing signal phasing and timing plans, understanding travel behavior, detecting incidents and managing response strategies, discovering and predicting travel trends, and sharing critical information in network users in real time.

To better understand the opportunities to realize these benefits, the research team in collaboration with TxDOT seeks to understand the capabilities, applications, and tools that enable the utilization of AI. The research team and TxDOT have developed a project plan and timeline that begins with a knowledge discovery phase and the development of this chapter and literature review, which will lead to the development of a Prospectus of AI in Transportation. The literature review presents a broad overview of subtopics related to the application of AI in transportation and is organized into the following sections: 1) the state of AI, 2) transportation data sources, 3) strategic initiatives, 4) applications of AI in transportation, and 5) technical tools and platform for AI applications.

Information from the literature review was used to form the basis of the Prospectus of AI in Transportation. The Prospectus will serve as a reference for transportation system owners and operators when exploring AI solutions to transportation network challenges, selecting software tools to be used in-house, or evaluating commercially available platforms. It will also be of assistance when assessing potential benefits of collecting and archiving transportation data. The latter may also be of interest to information and technology teams, which may use the Prospectus to inform some decisions related to data sharing and accessibility, and potentially data services to be offered to public agency partners and across departments.

2.1. The State of Practice in Artificial Intelligence

Artificial intelligence can be loosely interpreted as a process to incorporate human intelligence into machines, and it is an entire field of study that encompasses simple logic to far more complex statistical deep learning. AI ranges from simple “if-then” statements that are hard-coded and compiled for performing very narrow tasks to complex statistical models mapping raw sensory data into symbolic categories (Copeland, 2016).

AI underlies many existing technologies, including virtual assistants and driverless vehicles. Two popular AI techniques include deep learning (DL) and machine learning (ML). Figure 1 shows the relationship among AI, ML, and DL. ML techniques train computers to parse and discover hidden patterns within data and make actionable predictions without explicit programming. DL

enables applications that require fast processing and discovery on big data sources such as video processing and autonomous driving. These techniques hold new potential to provide insights for understanding the systematic design of transportation systems and pinpointing key situations that have propagating effects on entire networks.

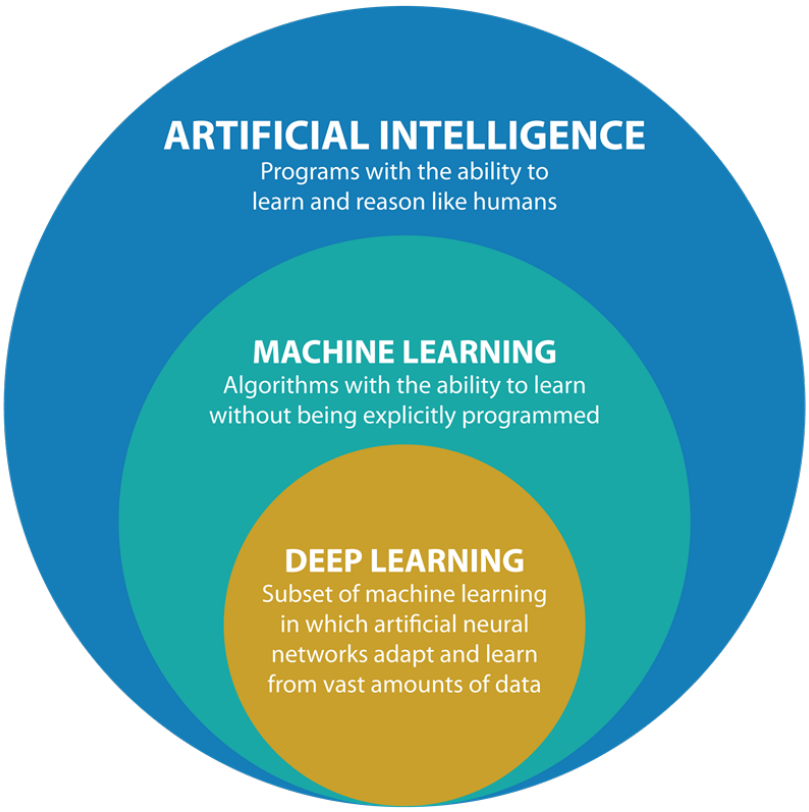


Figure 1. Relationship of AI, ML, and DL (Argility, n.d.).

Robotic process automation is typically rule-based and therefore a more primitive form of AI. It is realized in the form of business process automation software that can dramatically reduce the time spent on traditional workflows in a user interface such as creating or updating accounts, appointments, or other essential but tedious tasks. A promising emerging application of robotic process automation could encompass data restructuring, as increasingly complex but highly repetitive tasks can be automated. Another prominent subclass of AI contains the branches of natural language processing and natural language generation. These functional classifications rely on some combination of human speech recognition, natural language understanding that extracts information from written text, natural language generation that can produce narratives, and speech synthesis that speaks or reads out text. Advances in this branch of AI provide the backbone for simple, rule-based chatbots that can provide generalized responses to public inquiries, as well as more intelligent virtual agencies that can contextualize human questions and responses and respond specifically and appropriately. Computer and machine vision, a relatively computationally expensive class of AI, relies on images and video

captured by cameras. Algorithms to process these images enable outcomes such as object detection. For instance, traffic counts through an intersection can be estimated by identifying and tracking vehicles' trajectories through that intersection. Similarly, sound classification and audio recognition require more computational power on the back end and the installation of mobile or fixed sensors. However, such infrastructure can enable unprecedented quality of environmental monitoring, automatically identifying and classifying emergency or unusual events with new accuracy and speed. A synergistic but independent branch of AI is predictive and prescriptive analytics. This branch is built upon statistical algorithms that are becoming increasingly accurate at forecasting the probabilities of an array of physical, behavioral, and economic outcomes. At the predictive level, this can enable predictive maintenance and more cost- or time-efficient scheduling of resources to achieve the same or better outcomes in long-run infrastructure conditions. At the prescriptive level, a vast array of scenarios can be evaluated under different policy and operational decisions in order to assist decision-makers in identifying a context-sensitive, feasible course of action.

Advanced approaches to analyzing and understanding data, in order of increasing specificity, are ML, DL, and artificial neural networks (ANN). These fields and models underpin all of the most advanced realizations of the AI classifications described above. Computer scientists use these types of models and enormous quantities of data to observe and classify features of physical objects, the built environment, or spatiotemporal patterns and processes, and abstract them to recognize when a new data input is not identical to previously observed ones, but of the same class. The new level of abstraction achieved algorithmically is what enables counting vehicles, which are not identical in shape or size, understanding speech or text written by different individuals, or interpreting and assigning risk to different environmental signals.

More details on the capabilities and identification of the overlap between ML and DL are provided below.

2.1.1. Machine Learning

ML methods involve programing computers to learn semi-autonomously from data inputs in such way that they can make inferences and produce highly accurate predictions. Some of the most common ML techniques are described in the following paragraphs.

Supervised learning involves the use of human input—often in the form of classified images—to "teach" a machine how to identify or treat certain objects like images or words. For example, a human might label a string of images as "car" if the series contains a car and "not a car" if it doesn't contain a car. The computer can then analyze an image and guess between car and not a car and learn immediately if its strategy resulted in a success. This process allows the machine to learn how to change the way it analyzes images. Not all supervised examples have such clear

definitions of success, however; DL and regression scenarios can involve far more complex numerical definitions of success, depending on what the machine needs to learn.

- Classification algorithms label data records to categorize them into different classes.
- Regression algorithms return a numerical target or quantity for each data record.

Transportation application examples include predicting mode choice using non-parametric decision tree methods (Bhavsar, Parth, et al., 2017), as well as predicting link speeds using ANNs (Bhavsar, Parth, et al., 2017), or the number of trips that will originate or end in a particular place (Rodrigues, Francisco. & Ben-Akiva, 2013).

Unsupervised learning is a class of algorithms that take a set of data that contains only inputs and finds structure in the data, like grouping or clustering of unlabeled data points. Examples of unsupervised learning include:

- Clustering or automatically separating data set into groups according to similarity.
- Anomaly detection or automatically discovering unusual or unexpected data points or events.
- Association mining, which identifies sets of items or events that frequently occur together.
- Latent variable analysis, which is often used for data preprocessing to reduce data dimensionality or decompose the data set into multiple components.

Transportation application examples include clustering roadway segments or drivers into typologies based on similar characteristics (Júnior et al., 2017) or finding associations in driver behaviors and environments with crashes (Nandurge & Nagaraj, 2017).

Reinforcement learning (RL) is an area of ML concerned with how software agents take action in an environment so as to maximize the notion of total reward. This method directs unsupervised ML through rewards and penalties and is often modeled as a Markov Decision Process.

Compared to unsupervised learning, RL is different in terms of objectives. While the objective of unsupervised learning is to find similarities and differences between data points, the objective of RL is to find a suitable action model that would maximize the total cumulative reward of an agent.

One such useful transportation application example is the use of an RL-based multi-agent system for network traffic signal control where researchers have designed a traffic light controller to minimize delay and improve congestion in a five-intersection corridor. Although it was designed and tested only through a simulated environment, this method shows more

promising results compared to traditional methods of signal control design and illustrates the potential uses of multi-agent RL in designing traffic system and corridor management (Arel et al., 2010).

2.1.2. Deep Learning

DL is the most recent advancement of ML, and it can be applied to supervised, unsupervised, and RL. A differentiator of DL is its role in processing high dimensionality data through automatic feature selection. These capabilities enable video, audio, and text analysis in high dimensions. The following paragraphs provide an overview of current use cases of DL in transportation applications.

Transportation network representation: The optimal representation of constrained spatial networks that depicts dependent structures and flows can be aided by AI. DL methods have been applied to transportation network representation challenges by modeling spatial and temporal dependencies to represent the traffic conditions on links across a network in future time intervals (Ma, Yu, Wang, et al., 2015).

Travel choice prediction: Wang and Zhao implemented a deep neural network (DNN) for a discrete choice model that could be extended to travel mode choice, travel frequency, travel scheduling, destination and origin choices, and route choice problems. Results from their analysis showed that DNNs can out-perform traditional discrete choice methods. With DNN's high prediction accuracy, these methods are recommended when accuracy is of highest importance; however, it is cautioned that DNNs were shown to lose interpretability (Wang & Zhao, 2018).

Traffic flow prediction: Traffic flow prediction is a common goal in transportation modelling that aims to estimate the number of vehicles on a road segment for several time intervals in the future. Promising DL applications in this area include using DL techniques to predict traffic flows on links at future states, cluster dependent links, and cluster dependent flows (Huang, Hong, et al., 2014; Jin & Sun, 2008).

Traffic signal control: Increasing the efficiency of corridor management systems, including traffic signal controllers, can help to optimize vehicle flow in order to reduce congestion and emissions. Applications of RL have helped to design traffic signal control with limited data input. Deep ANNs have been shown to add capabilities to the RL method by leveraging big data to build adaptive signal control agents. The agents in RL combined with abilities of DL to handle big data have enabled a method that can more efficiently develop optimal traffic control policies (Liang, et al., 2018)

Automatic vehicle detection through video analytics: Convolutional neural networks (CNN), an application of DL, have been successfully used to identify objects in video data streams (Wang et al., 2015). Automated video detection is currently available in ITS devices, which often require dedicated hardware. The Performing Agency has participated in research that leverages video data streams from traffic monitoring cameras to estimate metrics of vehicular performance and pedestrian safety (Huang et al., 2017) using CNNs and custom algorithms for object tracking. Jidong et al. (2018) explore using a DL approach for the study of traffic safety at signalized intersections.

Travel demand prediction: Researchers have proposed that deep multi-view spatial-temporal network analysis could be conducted with DL to capture the spatial-temporal correlation of taxi demand in New York. This methodology could be extended to ride-hailing and ride-share services for predicting demand based on data on recent demand, drop-offs, weather, time, and area of city. Other adaptations have been used for predicted transit service demand based on historical and near-real-time data (Yao et al., 2018).

Traffic incident processing: Understanding the main cause of incidents and their impact on a traffic network is critical for a modern transportation management system. DL for traffic incident processing holds potential to help design response strategies for incidents, helping address network disruptions. DL has been applied to network and incident data to develop real-time network incident risk maps and predict certain types of collisions (Chen et al., 2017).

2.1.3. Evolving AI

AI in concept and implementation has been around since the mid-parts of last century. Subfields like ML and DL have seen major advancements in the past few decades. While today's AI is miles ahead of what it used to be, it will continue to evolve along with advances in computing, hardware, telecommunications, and more. With advancements, AI is set to become more powerful and efficient. Some predictions about the future of AI include the notion that AI models will become more like programmatic software with less human intervention for calibration and present as more blended algorithmic and geometric modules (Chollet, 2017).

2.2. Transportation Data Sources

When it comes to building a successful AI or ML system, data is as important as algorithm or model code. While higher processing power can train machines faster, it won't change the quality of results. Data is the foundation of ML, and the lower the quality of the data for training, the worse the result of the AI is. On the other hand, the better quality the data, the better the result can be. While AI can be applied to any data type, the specific type and amount that is required is highly dependent upon the problem that is being solved. Data management

and utilization strategies need to consider items such as relevancy, formatting and interoperability, accessibility, ownership, and more as the amount of data grows.

Globally, the amount of new data generated each year is growing exponentially; 90 percent of all of the world's data has been created in just the last two years alone (Marr, 2018). Some experts are now advising that that connected vehicles will be the most important new digital platform in the short term with 220 million on the road by 2020 (Stack, 2018). Equipment manufacturers, like Hitachi, have estimated that one connected vehicle will send 25 gigabytes of data to the cloud every hour (Hitachi, 2018). According to Cisco Systems, Internet-of-Things (IoT) devices produce 5 quintillion bytes of data every day and more than 30 billion connected IoT devices are expected to be online by 2020 (Stack, 2018). Finally, autonomous vehicles are expected to utilize 40 terabytes of data for every eight hours of operations (Winter, 2017).

In transportation, advances in the information, communication, and technology sectors have increased the availability of transportation data for both public and private entities. Data streams may be available to transportation owners and operators either through direct collection or partnership. Data streams can originate from sensors on two primary sources: either from fixed infrastructure sensors or mobile sensors inside a vehicle or mobile device. Sensor data can further be broken into three categories, further described below, relating to passive, semi-passive, and active data capture.

Other sources of data are useful when cross-examined with the data sources listed above. For example, incident reports, events schedules, weather data, or construction and lane closure plans may be mined and analyzed along with transportation sensor data. Furthermore, the implementation of high-resolution data logging in traffic signal controllers has made high-quality data available on signal phasing and performance, which can be useful for cross-validation of flow data on signalized corridors.

2.2.1. Passive Sensors

Passive sensors do not require direct communication with a vehicle. Examples include inductive-loop detectors, radar, camera or computer vision, and license-plate readers. These sensors are typically used to obtain information on traffic counts, lane utilization, speed, vehicle type, and speed. Table 1 provides a description of infrastructure sensor data from passive sensors.

Table 1. Data Elements from Passive Sensors

Inductive Loop Detectors	Roadway sensors that capture vehicle counts, speed, lane occupancy
Microwave Radar	Mounted radar sensors to detect vehicle presence, speed, and capabilities to detect pedestrians
Magnetometers	Provide information on traffic volume, lane occupancy, speed as well as vehicle length
Video	Use of video analytics, license plate recognition, and infrared technology for detection with some systems able to determine vehicle type, color, license plate number, intersection turn movements, near-collisions, etc.
Environmental Sensors	Laser, infrared, and video sensors, which are mounted on infrastructure to detect weather events such as rain, fog, and ice or snow, as well as high and low temperatures, and to measure visibility
Infrared sensors (passive)	Detection of energy that is emitted or reflected from vehicles, road surfaces, and other objects measuring speed, vehicle length, vehicle counts, and occupancy
Infrared sensors (active)	Vehicle passage, presence, speed, and vehicle classification information and infrared can be used for safety purposes to detect overheating vehicles or fire
Acoustic Detectors	Acoustic sensors can be used to measure speed, volume, carriageway occupancy, and presence

2.2.2. Semi-Passive Sensors

Semi-passive sensors require participation from drivers' devices without their knowledge and can include applications such as roadside Bluetooth®, Wi-Fi, tire-pressure sensor sniffers, radio-frequency identification (RFID), cellular hand-off signals, and applications on smartphones. Table 2 provides a description of infrastructure sensor data from semi-passive sensors.

Table 2. Data Elements from Semi-Passive Sensors

Bluetooth	Include capabilities for link-link journey tracking and data on average speed, travel time, route choice, and trips trend data
Wi-fi	Similar to Bluetooth capabilities and often used to detect transit traveler wait times and travel times
RFID	RFID has many applications and can provide detailed information for truckling, airport, tolling, and rail applications
Cellular handoff	By monitoring usage of cellular handoff data, approximate locations can be inferred, and route choice can be estimated

2.2.3. Active Sensors

Active sensors require that travelers participate knowingly and typically have a personal interest in providing data. Some example applications for these sensors include fleet-tracking networks (such as automatic vehicle location) and mobile applications that allow users to report scenarios such as accidents and construction. Table 3 provides a description of infrastructure sensor data from passive sensors.

Table 3. Data Elements from Active Sensors

Automatic Vehicle Location	Used to determine the geographic location of a vehicle by placing a GPS electronic device on-board a vehicle (typically a fleet vehicle)
Mobile Devices and/or Mobile Apps	Gathers information from applications that users elect to install and use from their mobile devices. Typically used for navigation purposes, to receive real-time information, find and secure a travel reservation, finalize a transaction, or track deliveries. Typically, a user location needs to be shared through GPS. Crowdsource apps can gather information on events such as traffic jams, incidents, police presence, etc.

2.2.4. Non-Sensor Data (Contextual Data)

Non-sensor, contextual data can be combined with data from the categories above to enhance insights and analysis. This data is important for providing a broader understanding of specific types of information and placing them in a larger picture—for example, using social media posts to understand transportation network events. While this data may be systematically collected by transportation (and transportation-related) agencies, it may not easily be structured or digitized. Table 4 shows examples of useful non-sensor data.

Table 4. Example Data Elements from Non-Sensor Data

Crash Records	Includes descriptive information about incidents, including locations, surrounding environment, severity of injuries and fatalities, estimated speed, estimated trajectory, and other relevant factors
Construction & Lane Closures	Include information on number of lanes closed, location (mile marker beginning and end), type of work being formed, duration, contact information, and more
Social Media	Includes sentiments about current network conditions, level of services, identification of special events, and more
Weather Event Data	Includes information on temperature, wind speed, weather events such as heavy rain or floods, air quality, and more

2.3. Strategic Initiatives

A limited number of strategic initiatives are currently underway to bring together transportation stakeholders to advance the state of knowledge and practice of AI in transportation. These few initiatives that primarily focus on data management are summarized below.

USDOT's ITS Joint Program Office (ITS-JPO) Enterprise Data Initiatives focus on enabling effective data capture from ITS-enabled technologies, including connected vehicles (passenger, transit, and commercial vehicles), mobile devices, and infrastructure, in ways that protect the privacy of users (ITS JPO, n.d.). Key goals of the initiatives include:

- Enable, develop, and share data visualization techniques and tools
- Develop traffic analysis and management techniques that take advantage of crowd-sourced data
- Coordinate the operation of mobile devices to be carried by transit passengers, which are themselves generating connected vehicle messages
- Develop techniques, such as dynamic interrogative data capture, that will reduce the amount of data that needs to be stored

USDOT's Safety Data Initiative was developed to advance integration of existing data sources and new sources of big data, advance data analytics, and create data visualizations to help policymakers with safety solutions. Since the inception of the Safety Data Initiative, USDOT has worked on several pilot projects that explore the possibilities of using data for safety applications. Examples include exploring the opportunity to estimate police-reported traffic crashes in near-real time by combining crowdsourced crash data from Waze with crash data

provided by the State of Maryland via the National Highway Traffic Safety Administration's Electronic Data Transfer pilot.

IEEE's Big Data Governance and Metadata Management (BDGMM) was developed under the IEEE Big Data Initiative, an effort funded by the IEEE New Initiative Committee to lead efforts in big data standardization. The program goal is to enable data integration and fusing among heterogeneous data sets from diversified repositories through a machine-readable and actionable standard data infrastructure (IEEE SA, n.d.).

2.4. Applications of AI in Transportation

Results of the literature review and scan of industry activities have been organized into the following six categories: 1) system and service planning, 2) asset management, 3) system operations, 4) communication and information, 5) business administration, and 6) public safety and enforcement.

2.4.1. System and Service Planning

This includes a comprehensive consideration of service and infrastructure development scenarios and strategies, such as determining transit routes, added capacity needs, signal phasing and timing plans, funding allocation, etc. Application examples include using sensors and telecommunications to determine route choice, travel behavior, origins and destinations, transit wait times, etc. Below is summary of applications and use cases of AI in transportation system and service planning.

Telecommunications call data record (CDR) for transit service planning: In partnership with the City of San Francisco, AT&T is tracking and analyzing CDR to determine trip-making patterns. The data reveals activity centers of work, shopping, and recreation, tracking when users go there and via which routes. This information is being used by the transit authority for service planning (Morris, 2015).

Wi-fi and Bluetooth tracking for transit wait times, station arrivals, and origin-destination (OD) determination: An experiment in New York City is looking at Bluetooth and wi-fi data to enhance public transit service by analyzing pings from cellphones and other devices to such sensors.

Air quality forecasting: IBM researchers have developed an ML system that can analyze data about pollution levels in Beijing to forecast changes to air quality up to 72 hours in advance. The system's forecasts are 30 percent more accurate than traditional predictions and have a resolution of one kilometer. The researchers are further developing the system to forecast

hypothetical scenarios, such as changes in the number of drivers on the road, and to extend the forecasts up to 10 days in advance (Knight, 2015).

Call data records for OD planning: AT&T and the City of Rio de Janeiro, Brazil, are analyzing big data for transportation system planning, and in one use case are using data from user CDRs for travel analysis within the city of Rio de Janeiro. Call records belong to the increasing set of "passive data" information sources that are generated by daily transactions performed by ordinary people and have applications beyond their original use. Each time a call is made on a mobile phone, the mobile tower associated with that call—usually the closest tower—is recorded for billing purposes. CDRs provide information on location, time, and duration of every call, generating huge databases that are being used to produce a citywide OD matrix at a lower cost than the traditional OD survey methods (Mehndiratta and Alvim, 2014).

Route choice and system utilization monitoring: Verizon is using a combination of CDR analysis and a proprietary ML algorithm to develop unique roadway signatures of city streets to more precisely determine vehicle route choice and real-time roadway link performance.

Driver Behavior Profiling: Júnior et al. (2017) investigated different Android smartphone sensors and classification algorithms to assess which sensor and classification method combination enabled driver characteristic classification with the highest performance. Results show that specific combinations of sensors and intelligent methods allowed classification performance improvement. Authors compared the performance of four algorithms: ANNs, Support Vector Machines (SVM), Random Forest (RF), and Bayesian Network. As a result, authors found the top-five performing assemblies for each driving event type. Results showed that (i) larger time-window sizes perform better; (ii) Android gyroscope and accelerometer data are the best to detect driving events; (iii) as general rule, using all sensor axes perform better than using a single one except for aggressive left-turn events; (iv) RF is by far the best performing, followed by multilayer perceptron networks; and (v) the performance of the top 35 combinations is both satisfactory and equivalent, varying from 0.980 to 0.999 mean AUC values.

Mode choice and mode detection: Jahangiri and Rakha (2015) applied different supervised learning methods from the field of ML to develop multiclass classifiers to predict transportation modes, including driving a car, riding a bicycle, riding a bus, walking, and running. Supervised learning methods that were considered included K-nearest neighbor, SVMs, and tree-based models that comprise a single decision tree, bagging, and RF methods. For training and validating purposes, data was obtained from smartphone sensors that included device accelerometer, gyroscope, and rotation vector sensors. K-fold cross-validation as well as out-of-bag error was used for model selection and validation purposes. Several features were created from the data, and a subset was identified for prediction through the minimum redundancy maximum relevance method (mRMR).

Data obtained from the smartphone sensors were found to provide important information to distinguish between the various transportation modes. The performance of different methods was evaluated and compared and showed that RF and SVM methods were found to produce the highest prediction accuracy.

Transportation mode identification and real-time CO2 emission estimation: Manzoni et al. (2010) proposed and tested an approach to estimate the individual carbon footprint of trips on various modes. The authors developed an application that could be run on standard smartphones for long periods of time. Given that that application could run on existing platforms (smartphones) that are already widely adopted, the method could have the potential for unprecedented data collection of mobility patterns. Specifically, the application could estimate real-time CO2 emissions using inertial information gathered from mobile phone sensors. In particular, an algorithm automatically classifies user transportation modes into eight classes using a decision tree. The algorithm has been trained on features computed from the Fast Fourier Transform coefficients of the total acceleration measured by the mobile phone accelerometer. A working smartphone application for the Android platform has been developed and experimental data have been used to train and validate the proposed method.

2.4.2. Asset Management

Asset management includes using sensors and data analytics to gather and predict insights about infrastructure and vehicle assets, their management and utilization strategies, long-term expenditure forecasts, and business management processes. Application examples include automated systems to identify pavement conditions such as cracks, ruts, and potholes, and signage and striping conditions. Other applications include automated fleet vehicle diagnostics and predictive maintenance.

Infrastructure maintenance and condition monitoring: The Georgia Department of Transportation in collaboration with Georgia Tech developed an automated system to identify pavement conditions such as cracks, ruts, and potholes. The system can also catalogue and assess other roadside infrastructure such as signs. The system employs lidar, GPS, and cameras—as well as AI in the form of machine vision—to outfit a sensing vehicle that can rapidly assess pavement conditions as it drives down a highway or road (Tsai et al., 2017).

Real-time truck parking: Researchers at the University of Minnesota have developed a computer-vision system that analyzes truck parking lots along the highway to automatically detect when a spot is available and notify approaching truckers. Roadside parking for long-haul truckers is limited and often crowded, making it difficult for them to plan ahead for where they can sleep and increasing the risk of overtired drivers staying on the road longer than they should. The system analyzes videos of truck stops and can distinguish between open and

occupied spaces with 95 percent accuracy, which is more reliable than systems that rely on sensors embedded in pavement (Marshall, 2017).

Street Bump in Boston, Massachusetts: This innovative project by staff in the City of Boston has found a way to address roadway maintenance by strategically identifying potholes through a big data and crowdsourcing approach. Boston sought a proactive and more cost-effective way to maintain and fix the city's roads using smartphones to identify and locate roadway maintenance issues. The result is an app called Street Bump, which utilizes the accelerometers and GPS of a driver's smartphone to identify potential infrastructure quality issues and associated location as the user drives. Data from the app can be aggregated by the city to map both urgently needed fixes, as well as areas in need of long-term investment.

Real-time monitoring and maintenance of full transit assets: Yarra Trams, the largest tram system in the world, is looking to big data, cloud computing and storage, mobile devices, and advanced analytics to transform its services. The agency is aiming to dramatically improve service on its 250 kilometers of double tracks by reconfiguring routes on the fly, pinpointing and fixing problems before they occur, and responding quickly to challenges, whether it's sudden flooding, major events in the city, or simply rush hour traffic. Yarra Tram's data-driven system works by tracking every one of the 91,000 different pieces of equipment that make up the tram network—from tram cars to power lines to tracks—using intelligent sensors and information from employee and passenger reports about service and equipment. For example, an automated wheel-measuring machine built into the track at the tram depot detects the condition of a tram's wheel when it rolls over it. The abundance of insights is collected and hosted on the cloud, where analytics are applied to help the Yarra Trams' operations team quickly respond to, prioritize, and coordinate maintenance and pinpoint future problems. As an example, data analysis identifies trends or patterns in tram and infrastructure repair history, using them as a guide for scheduling predicted maintenance. Maintenance crews receive work orders remotely on mobile devices, tackling repairs and potential disruptions before service is delayed. An app, meanwhile, gives passengers the latest information about track tram arrival, departures, or delays and alternative routes.

2.4.3. System Operations

Using technology and data to inform and automate strategies can optimize the safe, efficient, and reliable use of infrastructure for all modes. Application examples include intersection monitoring via camera, sensors, and telecommunications for conflict warning, pedestrian detection and notification, level-of-service monitoring, dynamic signal timing, and emergency response.

Network traffic signal control: Arel et al. (2010) introduced a novel use of a multi-agent system and RL framework to obtain an efficient traffic signal control policy. This approach aimed at minimizing the average delay, congestion, and likelihood of intersection cross-blocking. A five-intersection traffic network was studied in which each intersection was governed by an autonomous intelligent agent. Two types of agents, a central agent and an outbound agent, were employed. The outbound agents scheduled traffic signals by following the longest-queue-first (LQF) algorithm, which has been proven to guarantee stability and fairness, and collaborating with the central agent by providing it local traffic statistics. The central agent learned a value function driven by its local and neighbors' traffic conditions. The methodology proposed utilized the Q-Learning algorithm with a feedforward neural network for value function approximation. Experimental results demonstrated the advantages of multi-agent RL-based control over LQF-governed isolated single-intersection control, thus paving the way for efficient distributed traffic signal control in complex settings. Careful integration of the adaptive RL system with static LQF-based controllers was studied as both a developmental process and a gauging mechanism for indicating the degree of improvement that can be expected. Performance improvement was observed with respect to both average vehicle delay and cross-blocking likelihood, particularly in the context of high traffic scenarios.

Intersection monitoring, operations, and safety: Flir's TrafiRadar uses a Doppler radar and a camera to monitor vehicles at and approaching intersections to predict conflicts and traffic incidents. Using the machine vision enabled by both apparatuses, this product can actuate signal phase changes to mitigate predicted conflicts or alter phase timing in response to real-time traffic demand (ITS International, 2016). Miovision, a transportation analytics company, provides a cloud-based traffic management system that uses machine vision outfitted at intersections to measure traffic flows and intersection level of service, and detect near-miss traffic incidents to efficiently identify high risk intersections (Galang, 2017).

Intelligent streetlights: The city of Jacksonville's intelligent streetlights collect and analyze real-time usage data. Cameras connected to the lights can track traffic and pedestrian movements and decide when to dim or brighten each lamp. Sensors in the lights connected to a "smart parking" application can alert citizens to available parking spots—or even warn them when their parking meters are running out. This program is run in partnership with the GE Intelligent City Initiative (Cope, 2015).

Optimal work-zone operations: The operator of the Hong Kong subway system implemented cognitive technologies to automate and optimize the planning of engineering works. The planning system encodes rules of thumb learned by experts over years of experience, plus constraints such as schedules and regulations about maximum noise levels allowed at night. It employs a genetic algorithm that pits many solutions to the same problem against each other

to find the best one, producing an optimal engineering schedule automatically and saving two days of planning work per week. Though it automates the work of experts, it doesn't replace them (Shatsky, 2019; Hodson, 2014).

Dynamic Signal Control: Liang et al. (2018) proposed a solution to a traffic light control problem using the deep RL model. Data was gathered from vehicular networks and utilized information on two states: the vehicles' position and speed. Traffic control actions were modeled as a Markov decision process with a reward set on the cumulative waiting time difference between two cycles. Authors used a double dueling deep Q network (3DQN) with prioritized experience replay. The model proved to learn a good policy under both the rush hours and normal traffic flow rates with results showing a 20 percent reduction of the average waiting timing from the starting training. The proposed model also outperformed others in learning speed, which is shown in extensive simulation in SUMO software and TensorFlow.

Intersection safety and object identification: The Georgia Department of Transportation (GDOT) along with researchers from Kennesaw State University conducted a study into monitoring and assessing traffic safety at signalized intersections using live video images (Yang et al., 2018). The research team used DL for multiple object detection and tracking to explore and test the domain of traffic conflict monitoring and assessing. As a result, an AI-enhanced computational system was developed to automate the detection and quantification of traffic conflict events as they occurred in real time using traffic monitoring cameras already installed by transportation agencies. In alignment with the GDOT Vision Zero program, the research team developed surrogate safety measures such as time-to-collision, post-encroachment time, potential time to collision, difference in speeds (DeltaS), initial deceleration rate of the second vehicle, the maximum deceleration of the second vehicle (MaxD), difference in velocities (DeltaV), and safe stopping distance. To select the appropriate neural network architectures for the detection task, accuracy and speed were considered. Vehicles were detected using a DNN (i.e., Single Shot MultiBox Detector architecture) and tracked by an intersection-over-union-based method, which required minimal computation resources. Based on observed and predicted trajectories of vehicles in the study, a novel method was proposed to detect and quantify conflict events. Although the study has been focused on identifying and tracking vehicles at major signalized intersections, where a CCTV camera is typically installed for traffic monitoring, the proposed method can be generalized and employed at any locations with assistance of a high-resolution camera. Tensorflow Object Detection API (Huang et. al., 2017) was used for implementation.

Traffic flow prediction: Huang et al. (2014) proposed a DL architecture to predict future traffic flows on roadway links. The proposed framework consisted of two parts: a deep belief network (DBN) at the bottom and a multitask regression layer at the top. A DBN was employed for

unsupervised feature learning to find effective features for traffic flow prediction in an unsupervised fashion. This method has been examined and found to be effective for many areas, such as image and audio classification. To incorporate multitask learning (MTL) in the deep architecture, a multitask regression layer was used above the DBN for supervised prediction. The authors investigated homogeneous MTL and heterogeneous MTL for traffic flow prediction. To take full advantage of weight sharing in the deep architecture, the authors proposed a grouping method based on the weights in the top layer to make MTL more effective. Experiments on transportation data sets showed good performance of the deep architecture. Experiments showed that the approach achieved close to 5% improvements over other state-of-the-art methods. It was also presented that MTL can improve the generalization performance of shared tasks. These positive results demonstrate that DL and MTL are promising in transportation research. Two data sets were used in this study. One benchmark data set was the California PeMS data set, which collects information from inductive loops for more than 8100 freeway locations throughout the State of California. The other data set was from the highway system of China (the entrance–exit station of a highway—EESH). In each EESH, there was a station for charging and recording related information. From experiments on these two real traffic flow data sets, authors demonstrated that the proposed deep architecture can improve the accuracy of traffic flow prediction. With limited prior knowledge, it can learn effective feature representations. Both homogenous MTL and heterogeneous MTL can improve the generalization performance. The result of the approach can outperform the state-of-the-art approach with near 5% improvements in prediction accuracy.

Traffic flow prediction: Wu et al. (2018) implemented a DNN-based traffic flow prediction model (DNN-BTF) to improve the prediction accuracy for estimating traffic flow. The DNN-BTF model makes full use of weekly/daily periodicity and spatial-temporal characteristics of traffic flow. Inspired by recent work in ML, an attention-based model was introduced that automatically learned to determine the importance of past traffic flow. The CNN was also used to mine the spatial features and the recurrent neural network (RNN) to mine the temporal features of traffic flow. We also showed through visualization how the DNN-BTF model understands traffic flow data and presents a challenge to the conventional thinking about neural networks in the transportation field that neural networks represent purely a “black-box” model. Data from open-access database PeMS (described above) was used to validate the proposed DNN-BTF model on a long-term horizon prediction task. Experimental results demonstrated that this method outperformed the state-of-the-art approaches and that DL-based traffic flow prediction still needs more studies for finer-tuning.

Travel demand and congestion detection: Ma et al. (2015) extended DL models into large-scale transportation network analysis for predicting congested links in future states. Congestion was determined by setting a minimum speed threshold. A deep Restricted Boltzmann Machine and

RNN architecture was utilized to model and predict traffic congestion evolution on a network-based on GPS data from taxis trips. A numerical study in Ningbo, China was conducted to validate the effectiveness and efficiency of the proposed method using 4000 GPS-equipped taxis traveling from April 13, 2014, to May 9, 2014. Results showed that the prediction accuracy could achieve as high as 88 percent within less than six minutes when the model is implemented on a graphics processing unit (GPU)-based parallel computing environment. The predicted congestion evolution patterns were visualized temporally and spatially through a map-based platform to identify the vulnerable links for proactive congestion mitigation. The algorithm was implemented using Python Theano and was executed on a desktop computer with Intel i7 3.4GHz CPU, 8GB memory and NVIDIA GeForce GTX650 GPU (2GB RAM). Takeaways included that the overall algorithm prediction performance improved as the data aggregation level increased due to less data fluctuation and as the modeled time interval becomes longer. Data aggregated to the one-hour level led to 95 percent accuracy and only 43 percent of congested links could be correctly predicted when the aggregation level was set at 10 minutes. This implies that the data aggregation level influences the prediction outcomes and should be carefully selected.

2.4.4. Communications & Information

This includes using ML and natural language processing to transform unstructured data into structured data or mined insights. Application examples include scanning Twitter feeds to target disaster and hurricane response efforts, identifying traffic patterns associated with dangerous road conditions or incidents, and forecasting dangerous levels of air pollution.

Social media analytics for emergency response: AI for disaster response has been implemented around the world to classify tweets during natural disasters. The ML algorithm employed sorts through thousands of tweets to deliver information about areas in need of aid and that have experienced damage and can bypass non-informative tweets based on user-specified criteria. This application can save officials and emergency responders crucial time and resources during disaster scenarios (AIDR, n.d.).

Social media analytics for incident detection: Zhang et al. (2015) proposed a hybrid mechanism based on latent Dirichlet allocation and document clustering to identify and mine incident-level semantic information, while spatial point pattern analysis was applied to explore the spatial patterns and to assess the spatial dependence between incident-topic tweets and traffic incidents. Authors assessed the potential of using processed social media for traffic incident detection. Twitter (now known as X) in Seattle, Washington, was chosen as a representative sample environment for this work. A global Monte Carlo K-test indicated that the incident-topic tweets were significantly clustered at different scales up to 600 meters. The nearest neighbor clutter removal method was used to separate feature tweet points from clutter; then a density-based algorithm successfully detected the clusters of tweets

posted spatially close to traffic incidents. In multivariate spatial point pattern analysis, K-cross functions were investigated with Monte Carlo simulation to characterize and model the spatial dependence, and a positive spatial correlation was inferred between incident-topic tweets and traffic incidents up to 800 meters. Finally, the tweet intensity as a function of distance from the nearest traffic incident was estimated, and a log-linear model was summarized. The experiments supported the notion that social media feeds acted as sensors, which allowed enhancing awareness of traffic incidents and their potential disturbances.

Assisted Intelligence in application sorting and approval: Hong Kong's Immigration Department was processing more than 4 million visa applications in 2004. The department processed about 100 different application forms related to visas, travel documents, identity cards, and other issues. To streamline the process, in 2007 the Hong Kong Immigration Department invested in developing an algorithmic system that sorted passport applications and was trained to classify applications into three broad categories: approved, denied, and a gray area. Once the algorithm classified visa applications, the system would transfer the data to a visa officer, who reviewed the documents to make the final decision (Desouza, 2018).

Translation services: AI translation startup Unbabel offers a translation service that uses a combination of crowdsourcing and ML to translate businesses' customer-service operations into 14 different languages substantially faster and cheaper than traditional translation services, which can make it easier for businesses to reach international audiences. Unbabel's algorithm automatically translates web pages, customer service emails and chats, and social-media posts for as little as $0.02 per word, and a team of human editors reviews the translations for grammar and consistency (Mehr, 2017).

Vehicle image classification and video querying: Huang et al. (2017) proposed a framework to process large-scale traffic data from monocular traffic cameras and an implementation of a prototype application to classify images. This prototype enabled users to conduct dynamic analysis tasks on the content of video feeds through a query interface. Candidate objects in each frame were detected and identified, and then these objects were merged and filtered based on their previous appearance to generate a trajectory of those objects throughout the video. This enabled structured data that could be processed and analyzed through novel database architecture—HiveQL backed up by Hive/Spark cluster. The scalability of the application was realized through a distributed object recognition and tracking algorithm implementation, using Hive/S-park programming models. Usability was demonstrated with two use cases: counting vehicles on an arterial street and identifying potential pedestrian-vehicle interactions with real video feeds. Experimental results proved to be encouraging, with vehicle counts being more than 85 percent accurate in the analyzed videos. Sample queries to identify

turning vehicles and measure the proximity of pedestrians to vehicles were presented and their performance validated. The results show the usability of the application for real-world problems. The proposed approach effectively transformed digital video data into queryable information. Thus, it formed a sharp contrast with the traditional analysis methods used in studying traffic camera video within the transportation research field. Most traffic management centers control several monocular traffic cameras that are often used only for monitoring purposes. This approach has the potential to enable new data workflows that require minimal human intervention to generate valuable data sets that complement information from other fixed and mobile sensors. This work demonstrated performance optimization and extending parallel computations from using GPU only to using multiple CPUs and many-core hardware.

2.4.5. Business Administration

Business administration applications use ML, data mining, and natural language processing and generation to optimize typical administrative functions, such as sorting, formatting, cleaning data, populating forms, scheduling appointments, and responding to routine public inquiries. Application examples include machine vision that reads handwriting and automates sorting, software that can manage complex scheduling tasks, and chatbots that can engage in and respond to simple requests and questions from the public.

Drafting of documents and contract templates: This type of application uses natural language generation (NLG) AI, which is already being used in dozens of newsrooms, including Bloomberg and the Associated Press, to mine data, create text for data sets, and write at a pace of 2,000 stories per second. In these scenarios, NLG can also help non-data-science employees better and more efficiently understand the data. Lawyaw is building software to automate the process of drafting similar but custom contracts by letting lawyers turn previously completed documents into smart templates (Tepper, 2018).

Virtual assistant: Startup X.ai offers a virtual-assistant service named Amy that can analyze employee calendars and emails to automatically schedule meetings and adjust calendar appointments. Users can copy Amy in emails when they want to set up meetings, and Amy will analyze email text to determine the topic and time of a meeting, determine if there are any conflicts, and automatically schedule calendar appointments. Amy can also search for and add relevant phone numbers, reschedule meetings by conversing with users, and learn users' preferences over time (Popper, 2016).

A**utomated industrial design generation**: Engineering software company Autodesk has developed computer-aided design and drafting software called Dreamcatcher that uses AI to automatically generate industrial designs based on a designer's specific criteria, such as function, cost, and material. Dreamcatcher can generate multiple alternative designs that meet

the same criteria and provide designers with performance data for each design, alter designs in real time based on designer feedback, and export finalized designs into formats used for fabrication (AutoDesk, n.d.). Workforce analytics company Kanjoya uses natural language processing and ML to analyze language used in the workplace, such as employee performance evaluations, interview notes, and office communications, to identify signs of implicit gender bias, so companies can treat employees more fairly. Kanjoya's software can identify potentially discriminatory or abusive language and emotions or intent in language, as well as identify biased decision-making, such as by revealing that "assertiveness" is often associated with positive reviews and promotions for men, but negative reviews for women (Giang, 2015).

Hardware and equipment performance, **alerts, and optimization**: Google has implemented AI software developed by fellow Alphabet subsidiary DeepMind in one of its data centers to automatically optimize energy efficiency while responding to factors such as increased usage and changing weather. The system constantly monitors 120 variables, including server usage and windows, and learns how to adjust equipment performance and cooling systems to run the data center as efficiently as possible. The system has reduced the data center's energy consumption by 15% (Clark, 2015).

Hardware and **cognitive predictive maintenance**: Analytics company DataRPM has developed a tool called Cognitive Predictive Maintenance that uses ML to monitor machine-component performance through networked sensors in real time to detect early warning signs that a machine might be breaking down to prompt preventative maintenance. Servicing machines before they break down is more cost effective than fixing them after they break, and DataRPM can predict when maintenance is needed 300 percent more accurately and 30 times faster than traditional methods, saving its customers 30 percent in maintenance costs (Progress, n.d.).

Data management: Sorting through dark data by using ML and combining its power with algorithms that address how to sort and handle different types of emails, documents, images, etc., stored on servers, ML, AI, and analytics can go to work on this disparate data and pre-sort it. Humans can then review what the automation recommends as a data classification scheme, tweak it, and perform the scheme. Part of the process could also address data retention, with the analytics producing a set of recommendations on which data could potentially be purged from files. Discarding un-needed or obsolete data by objectively identifying data that is seldom or never used and recommending what to discard (but it doesn't automatically discard and defers to discernment of employees). For instance, these processes can pick out pieces of data or records that haven't been accessed for more than five years, indicating that the data could be obsolete. This saves an agency time hunting down this potentially obsolete data, because now all employees need to do is to determine whether there is any reason to keep it.

Aggregating data: When analytics developers determine the kinds of data they need to aggregate for queries, they often produce a repository for the application, and then pull in various types of data from different sources to make up an analytics data pool. To do this, they must develop integration methods to access the different sources from which they pull data. ML can make this still very manual process more efficient by automatically developing "mappings" between data sources and the application's data repository reducing integration and aggregation times. Organizing data storage for best access using technology advances to enable IT departments to use "smart" storage engines that use ML to see which types of data are used most often, and which are seldom or never used. The automation can be used to automatically store data in fast or slow storage, based on the business rules inserted into machine algorithms. The automation saves storage managers from having to address storage optimization manually.

2.4.6. Public Safety and Enforcement

Public safety applications use advanced analytics, data mining, computer vision, and other innovations to enhance the safety and security of citizens. Applications serve the mission of keeping physical spaces and communities safe, but also our virtual ones. Applications will help direct police and emergency services units in the event of a crime or an accident, improve accuracy, and boost the rate of response.

Safe routes for hazardous materials transport: Matias et al. (2006) proposed a methodology to model the degree of remedial action required to make short stretches of a roadway suitable for dangerous goods transport, particularly pollutant substances, using different variables associated with the characteristics of each segment. Thirty-one factors determining the impact of an accident on a particular stretch of road were identified and subdivided into two major groups: accident probability factors and accident severity factors. Given the number of factors determining the state of a particular road segment, the only viable statistical methods for implementing the model were ML techniques, such as multilayer perceptron networks, classification trees, and SVMs. The results produced by these techniques on a test sample were more favorable than those produced by traditional discriminant analysis, irrespective of whether dimensionality reduction techniques were applied. The best results were obtained using SVMs specifically adapted to ordinal data. This technique takes advantage of the ordinal information contained in the data without penalizing the computational load.

Incident analysis: Nandurge and Nagaraj (2017) conducted a study to help determine the main factors associated with road traffic accidents, which is a critical objective of accident data analysis. Due the heterogeneous nature of road accident data, this can be a complex task. Partitioning (clustering) was used to help overcome heterogeneity of data and finding similarities among crashes. The method proposed utilized the k-means clustering method as the

main task of segmentation of road accident data. Further, association rule mining was applied to discover situations related with the occurrence of the whole data set and the occurrence of clusters recognized by the k-means clustering algorithm. The study used accident data from the Maharashtra (India) road network in 2015 and 2016. K-means clustering resulted in five clusters based on attribute accident type, road type, light condition, and road characteristics. Association rule mining was applied to every cluster as well as the whole data set to create rules (or common causes or characteristics to the incidents). Strong rules were used for analysis with high lift values. The rules of each cluster provide insights into situations related to incidents.

Traffic analysis during inclement weather: Koesdwiady et al. (2016) applied DL to traffic and weather data to improve traffic flow estimation during weather events. The study had two objectives: first, to investigate a correlation between weather parameters and traffic flow and, second, to improve traffic flow prediction by proposing a novel holistic architecture. It incorporated DBNs for traffic and weather prediction and decision-level data fusion scheme to enhance prediction accuracy using weather conditions. The experimental results, using traffic and weather data originated from the San Francisco Bay Area of California, validated the effectiveness of the proposed approach compared with the state of the art. Experiment results show that the data-driven urban traffic system prediction outperforms comparable state-of-the-art techniques. This higher traffic prediction accuracy ensures better operation and management traffic strategies.

Audio-recognition for safety analysis: One Llama Labs has developed a smartphone app called Audio Aware that uses ML to identify sounds associated with dangerous situations, such as sirens or squealing tires, and warn hard-of-hearing users about the noise. Audio Aware can identify a variety of dangerous sounds through a smartphone's microphone, as well as allow users to record and share their own, and when it detects one, it will play an amplified version of the noise through headphones, which can help partially deaf users stay more aware of their surroundings (Metz, 2014).

Disaster response: The Qatar Computing Research Institute has developed an open-source tool called Artificial intelligence for Disaster Response (AIDR) that uses ML to monitor and analyze Twitter posts and automatically compile Twitter activity related to a particular crisis to aid humanitarian response. In a test during the 2013 flooding in Pakistan, volunteers trained AIDR on tweets related to the crisis, and it could determine if new tweets were related to the Pakistan floods, based on their text, time stamp, and geotag, with 80 percent accuracy (Collins, 2013).

Explosive device detection: An EU-funded project called the Autonomous Vehicle Emergency Recovery Tool (AVERT) uses a system of four autonomous robotic platforms that coordinate

with each other to position themselves under a vehicle suspected of having an explosive device and move it to a safe location. AVERT uses sensor technology called LIDAR to map its environment and automatically develop an extraction route for suspicious vehicles in situations where it is too dangerous or difficult to use normal bomb-disposal tools (Moren, 2019).

Deep web analysis: The U.S. Defense Advanced Research Projects Agency's (DARPA) Memex program has developed a tool that scans pages on the deep web—websites that are not indexed on search engines—and analyzes their contents for signs of illegal solicitations of sex, which are often linked with human trafficking, to aid investigations. Because pages on the deep web can be difficult to access and navigate, they can be attractive covers for criminal activity. With Memex, authorities can gain new insights into specific investigations as well as broad trends about human trafficking, such as by generating heat maps of regions with a high density of illegal solicitation (Greenemeier, 2015). 13,

Identifying intoxicated individuals from cell phone data: Uber has developed technology using AI to identify levels of intoxication of potential passengers before picking them up. The system uses a computer model to identify user and trip characteristics indicative of the uncharacteristic user states. Historic data about past trips is used to train a computer model to predict intoxication of the user submitting a trip request. Ride-requester activities, such as how many typographical errors are made requesting a ride, the amount of time it takes for the person to interact after receiving new information via the mobile app, the angle at which the phone is being held, movement of the device during the request entry, or a user's travel speed can be factored in. The system then analyzes the data and generates a prediction about the state of the user, such as whether the person appears to be drunk. The system can compare past trip behaviors of the person to identify any deviation. Based on the results, the system could match the person with certain drivers and modify pickup or drop-off locations.

2.5. Methods and Tools

Tools, languages, libraries, databases, and architecture should be considered for optimal implementation of ML and DL. Each should be carefully considered, as each has implications for computation time, hardware needed, etc. This section provides an overview of libraries, platforms, and tools for AI implementation, as well as an overview of database and data access tools that support data ingestion and management. Finally, considerations on computing resources and environments will be provided.

2.5.1. Libraries, Platforms, and Tools

The following provides an overview of available libraries, platforms, and tools that can be openly accessed for the implementation of ML and DL algorithms.

2.5.1.1. Scikit-Learn

Scikit-learn is a free and open-source library for ML that works with NumPy, SciPy, and matplotlib within Python. The Scikit library contains tools that can be used for various ML and statistical modeling techniques, including supervised learning, cross-validation, unsupervised learning, and feature extraction. As Scikit is an open-source library, there is a large community involved in developing and refining its features, with around 35 contributors (Jain, 2015).

2.5.1.2. Tensor Flow

Developed as a library for numerical computation and large-scale ML applications, Tensor Flow is an open-source library created by Google. Using Python to build the application, it utilizes C++ to execute the programming. Tensor Flow can be used for DNNs, with applications of image recognition, word embeddings, RNNs, sequence-sequence models for machine translation, and natural language processing. The same models that are used for training can also be used for at scale prediction (Yegulalp, 2019).

2.5.1.3. Theano

Another open-source library, Theano is designed to be used as a compiler for mathematical expression within Python (Theano, n.d.). Working with NumPy and native code, Theano is designed to make code run as efficiently as possible. One of the first libraries to be developed, it is used as an industry standard for DL applications, such as large neural network algorithms (Brownlee, 2019).

2.5.1.4. Caffe

Caffe is a DL framework developed by Berkeley AI Research along with community contributors (Caffe (a), n.d.). Caffe can be used in either Python or Matlab, as both have interfaces (Caffe (b), n.d.). Caffe can be used to perform image classification and filter visualization, object detection, and neural networks, among other DL applications (Shelhammer et al., n.d.).

2.5.1.5. Keras

Built with the focus on fast experimentation capabilities, Keras is a high-level neural network API written in Python. With it being written in Python, it can run on top of other Python-based libraries (TensorFlow, CNTK, and Theano). Keras is designed to support both convolutional networks and recurrent networks, and combinations of the two, while providing easy and fast model prototyping through user friendly modules that can be easily added and modified to construct increasing complex networks (Keras, n.d.).

2.5.1.6. PyTorch

PyTorch is a Python-based library for scientific computing and is a preferred DL research library due to its speed and flexibility. Allowing for easy use and model building of complex neural networks, it can be used to run and test parts of the code in real time versus having to execute the entire code. PyTorch's functionalities can be extended with the use of other Python packages (NumPy, SciPy, and Cython) (Shetty, 2018).

2.5.1.7. Google ML Kit

Google ML Kit allows for ML within Android and iOS apps. ML Kit APIs can run on the cloud or run on-device, with cloud APIs offering higher levels of accuracy. Cloud-based APIs can perform text recognition, image labeling, and landmark recognition, whereas the on-device APIs provide applications like face detection, barcode scanning, object detection and tracking, language identification, translation, and smart reply, as well as the features that are available on the cloud-based APIs (Firebase, n.d.).

2.5.2. Databases & Data Access

When it comes to building a successful AI or ML system, data is as important as algorithm or model code. It's not only the volume of data or data quality that matters, although both are very important. It is equally critical to have efficient ways to manage data at scale, particularly for the special needs of ML, such as data versioning for training models, loading streaming data, and maintaining reliable event-by-event history, as well as a way to archive data for training and testing. The following provides an overview of database tools and data access and management tools.

2.5.2.1. Databases: Relational and NoSQL databases

Relational databases have been a foundation of modern computing over the past couple of decades. Examples of such databases include versions of SQL (Structured Query Language) MySQL, SQL Server, and SQLite. With the rise of the internet, networked systems, and big data, the quantity, scale, and rapidly changing nature of data being stored has not landed itself to easy storage in these traditional databases. Relational databases have struggled to adapt to the complexity and ever-changing nature of modern computing and its high volume of data. As a result, NoSQL (Not Only SQL) databases are replacing relational databases in many enterprises and modern applications as its design emphasizes non-relational data storage. Relational and NoSQL databases can be characterized and compared in many dimensions. See Figure 2 for such a comparison.

Relational databases are an organized set of data with pre-defined relationships. Data is organized into structured sets organized into related tables. Relational data can be accessed in many different ways without reorganizing the database tables themselves. SQL is the primary interface used to communicate with relational databases and became a standard of the American National Standards Institute (ANSI) in 1986. The standard ANSI SQL is supported by all popular relational database engines (mentioned above), and some of these have extension to support functionality that is specific to that engine. SQL is used to filter, add, update, delete (and otherwise manage) data.

NoSQL systems are non-relational, distributed databases that configured for large-scale data storage and for massively parallel, high-performance data processing across a large number of servers. These systems we created to meet the need for flexibility, performance, and scale, and to support a large variety of use cases, including real-time computation and analytics. These systems have origins in large internet companies that needed to manage high velocity data and to scale quickly.

Feature	NoSQL Databases	Relational Databases
Performance	High	Low
Reliability	Poor	Good
Availability	Good	Good
Consistency	Poor	Good
Data Storage	Optimized for huge data	Medium sized to large
Scalability	High	High (but more expensive)

Figure 2. Comparison of NoSQL & Relational Databases (Bispo and Andres).

Extract, Transform, and Load (ETL)

This shift to real-time processing generated a major change in system architecture: from a model based on batch processing (where a set of data is collected over a period of time an input in a "batch") to a model based on distributed message queues and stream processing (involving continual input). Apache Kafka (an open-source tool developed at LinkedIn) is one example of distributed message queue that has emerged for modern data applications to process real-time data feeds.

2.5.3. Computer Resources and Environments

During the traditional development phase of a programming project (developing, testing, and refining code) developers often program on their local computer, typically a desktop that runs on a non-server operating system (e.g., macOS, Windows, Ubuntu). Developers can also implement on a virtual machine, which is virtual/remote environment that behaves like an actual computer. Remote access, control, and computing is possible with technologies such as a remote desktop or virtual network machine. The benefit of remote development and computing is that it offloads the system memory (RAM), processing (CPU) requirements, and load to a more powerful machine. In instances of virtual machines hosted on cloud-based Infrastructure-as-a-Service (IaaS) platforms such as Amazon Web Services and Microsoft Azure, they can be easily configured to handle a large variety of memory, processing, and data storage requirements.

An additional benefit of remote development is that remote machines are loaded onto commercial servers, which means that they are running highly optimized operating systems that are meant for server-like usage, as opposed to personal desktop operating systems.

Remote development may be necessary in many cases when the data set being used is larger than the local machine's available system memory (RAM), or the processing requirements for a given task are greater than the local machine's capability. When limited by either RAM or CPU, it can result in extremely slow or impossible computations. If this happens, developers use a smaller subset of the data for local development and prototyping purposes.

Current trends in graphics processing units (GPU) design and configuration have enabled larger dedicated memory, higher bandwidth for graphics memory, and increased internal parallel processing. Increasingly, GPUs are designed with increasing degrees of programmability that extends to applications outside graphics processing.

GPUs can greatly accelerate the training process for deep learning models. Training models for tasks like classifying images, conducting video analysis, and natural language processing involves compute-intensive matrix multiplication and other operations that can take advantage of a GPU's massively parallel architecture. Training a deep learning model that involves intensive compute tasks on extremely large datasets can take days to run on a single processor. Developers can design programs to offload those tasks to one or more GPUs, thereby reducing training time from days to hours.

2.6. Conclusions

This literature review provides a background on artificial intelligence techniques and tools, and explores transportation applications in system and service planning, operations, asset management, public safety and enforcement, communications, and business administration. Researchers found that there is significant potential for the application of AI to help transportation system operators and owners to advance their goals. Existing applications range from less sophisticated algorithms with simple logic to cutting-edge DL methods. But even the most sophisticated algorithms cannot produce novel insights without well curated data. Our literature review includes a summary of typical AI data sources used in transportation and their challenges and limitations. The research team prepared a Prospectus of AI in Transportation to summarize the gathered knowledge (Product 1).

Chapter 3. Data Survey

This chapter provides an overview of transportation data available to TxDOT, focusing on data that may support enhanced transportation network operations, and integrated corridor management in particular. The goal of this data survey is to provide a systematic description of available information in order to facilitate the effective selection of use cases to be tested later during this project.

Since there is no single centralized repository of statewide data, we have organized this chapter based on known data access points/streams, including some that are available only to specific TxDOT districts.

This data survey also includes a separate description of two third-party data types, probe-based speeds and Waze for Cities Data, which are available to TxDOT to some degree. Some of the probe-based and Waze data is shared through the access points described earlier, but given the dynamic nature of the agreements between TxDOT and the data providers, we are not able to present a final description of coverage and access.

The data streams described in this data survey became known to CTR through previous research and implementation work for TxDOT, and by reaching out to individuals within TxDOT on the recommendations of project team members. This document is not expected to cover all data sources available to TxDOT.

3.1. Overview of Data Access Points and Data Streams

This data survey is centered on identifying versatile data sources to support transportation network planning and operations. CTR considered mostly datasets/data streams that are a) available or likely to become available statewide, and b) maintained and updated systematically. We considered datasets that are generated by TxDOT's Intelligent Transportation System (ITS) devices and other agency workflows, as well as proprietary datasets to which TxDOT has access. Table 5 presents the major data streams/access points to be discussed in this data survey.

Table 5. Data considered in Data Survey

Data Access Point	Access Type	Data Type
Lonestar	Stream	Traffic speeds/roadway network/traffic control/other.
TxDOT Data Lake	Repository	Data in Lonestar plus other infrastructure and general system data.
Highway Condition Reporting System	Stream	Roadway closures, incidents, traffic speeds.
Statewide Traffic Analysis Reporting System	Repository	Traffic volumes and vehicle classification.
Crash-Records Information System	Repository	Traffic incidents.
Third-Party Traffic Speeds and Travel Times	Stream/Repository	Traffic speeds and travel times.
Waze for Cities Data	Stream/Repository	Incidents, closures, location of traffic jams, other.
Transit Data	Stream/Posted Regularly	Transit routes, schedules, fares, vehicle location, ridership.
Experimental/Project-Specific Data	On Request	Traffic volumes, speed, weather conditions, flooding.
TxDOT Open Data Portal	Repository	Variety of geo-coded data.

The following sections provide additional information about each proposed data stream/access point. We also briefly discus additional data sources that are of interest and which may be further explored in the final version of this document pending feedback from the project team and the data owners.

3.2. Lonestar

The Lonestar system is an advanced traffic management system (ATMS) developed and maintained by the Southwest Research Institute (SwRI) on behalf of TxDOT. A fork of the software has also been deployed in Florida for FDOT under the label SunGuide.

The Lonestar system is mainly designed to share data from intelligent transportation systems in near-real time both within TxDOT, and between TxDOT and partner agencies/consultants. Lonestar data, provided in XML format, may be accessed programmatically using a center-to-center (C2C) application program interface (API). The API implements standards from the National Transportation Communications for ITS Protocol, or NTCIP.[1] A superset of the data types that may be communicated using the C2C protocol is presented in Table 6.[2]

[1] https://www.ntcip.org/center-to-center/
[2] Texas Department of Transportation. 2017."Center-to-Center Communications Status Interface Control Document - C2C-SICD-6.1.4.". Available from TxDOT.

Table 6. Lonestar XML data types

Network Data	CCTV snapshot Data	Parking Lot Data
Traffic Condition Data	HAR Data (Radio)	Rail Crossing Data
Incident Data	Ramp Meter Data	Bus Stop Data
Lane Closure Data	Traffic signal Data	Bus Location Data
Special Events Data	HOV Data	Rail Stop Data
Emergency Management Data	School Zone Data	Rail Location Data
DMS Data	Rev. Lane Data	Park and Ride Data
ICS Data	Dyn. Lane Data	Vehicle Priority Data
CCTV status Data	ESS Data (Weather)	Remote Message Data

Most of the data available through Lonestar is updated every few minutes/hours. Users may access the XML at appropriate time intervals to retrieve data in near-real time or to archive it for future use. Users may also request that any data stream broadcasted through Lonestar to be archived in TxDOT's Data Lake.

Data availability in Lonestar varies by district. As an example, the following data streams are available in the Austin District (among others):

- CCTV (closed-circuit television) snapshots and status
- DMS (dynamic message sign) data
- LCS (lane control signal) data
- Network data (e.g., roadway lengths and directions)
- Traffic conditions data from multiple sources, such as travel times on I-35 based on Bluetooth data

3.3. TxDOT Data Lake

The Data Lake is a data repository maintained by TxDOT's Information Technology Division (ITD), which may be used to maintain long-term access to data streamed in real time (such as that provided through Lonestar). Users may request for specific data streams to be archived in the Data Lake, and ITD can provide customized API data access to such datasets.

A data dictionary describing data currently available in the Data Lake is currently in development by ITD and will be incorporated into the final version of this document. As of February 2020, there are thirty-six databases in the Data Lake, spanning more than 1,200 tables.[3] These include C.R.I.S. data (Section 7), Lonestar C2C data (Section 3), and INRIX data

[3] Tentative list of data tables provided by Ross Alaspa on February 2020.

(Section 9). The latter has been archived since 2018 for selected freeways in Texas, and data gaps may be present due to communications issues.

3.4. Highway Condition Reporting System

The Highway Condition Reporting System (HCRS) provides real time information concerning traffic conditions on Texas highways. Drive Texas is a friendly user interface to HCRS data that displays conditions for roadways on the TxDOT-maintained system, including interstates, US highways, state highways, and farm- (FM) and ranch- (RM) to-market roadways (Figure 3).[4]

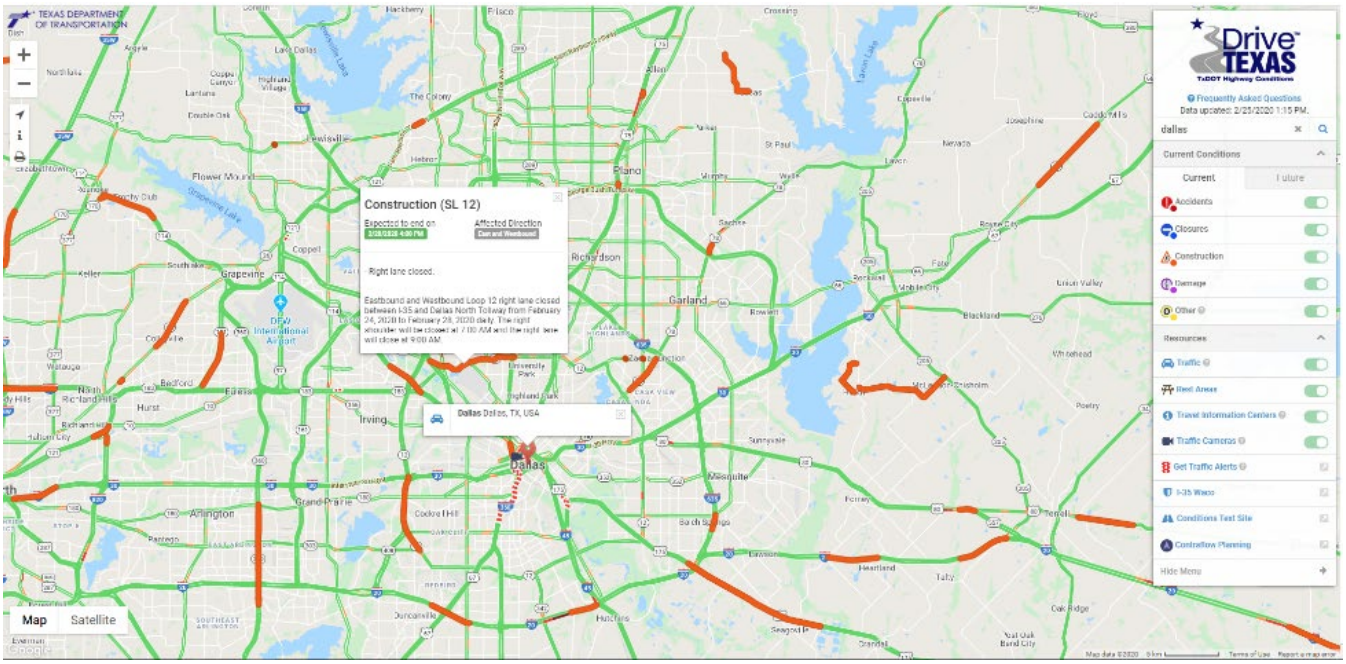


Figure 3. Drive Texas User Interface (http://drivetexas.org/)

TxDOT also provides API access to the HCRS on request.[5] The API is intended to support communications and inter-agency coordination during emergencies. Information is provided in GeoJSON, KML, and text (.csv) formats, which are popular data types that can be easily used for mapping reported conditions. A data dictionary for the HCRS is in development,[6] and further integration with other systems, including Lonestar (Section 3) is envisioned in the short/medium term.[7]

[4] https://drivetexas.org/faq
[5] https://api.drivetexas.org/
[6] As per communications with Sarah Berryhill on February 2020.
[7] As per feedback from James Kuhr on January 2020.

3.5. Statewide Traffic Analysis and Reporting System (STARS)

The Statewide Traffic Analysis and Reporting System (STARS) is used by TxDOT to comply with federal requirements for DOTs to develop, establish, implement, and operate a traffic monitoring system (TMS).[8]

Midwestern Software Solutions (MS2), the designer of the current STARS deployment, has implemented a software package called Traffic Count Database System (TCDS) to generate a data portal that allows for data exploration and download (Figure 4).

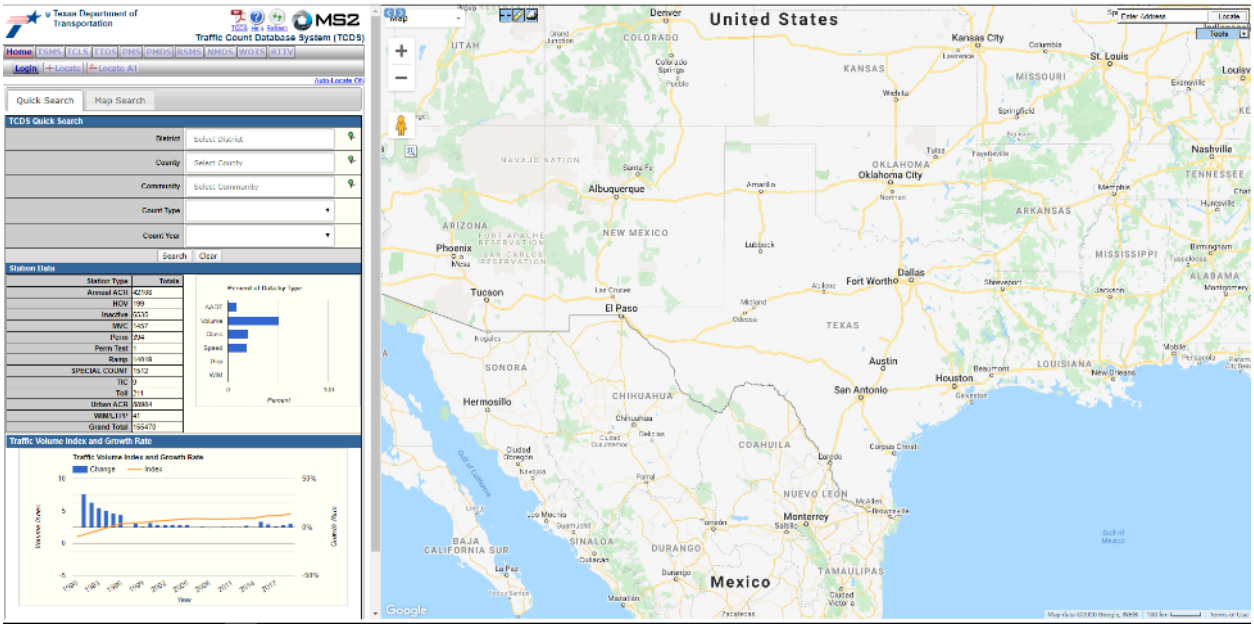


Figure 4. STARS II user interface.

The publication database, available since 2021, facilitates data querying and retrieval.[9]

The STARS system provides access to the following data, some of such data is collected at permanent count stations, which report data per lane and hour (Table 7).

- Pneumatic Tube Counts
- Manual and Automated Vehicle Classification Counts
- Weigh-in-Motion (WIM) Data

[8] https://www.law.cornell.edu/cfr/text/23/part-500/subpart-B

[9] As per communications with David Freidenfeld on February 2020.

TPP operates approximately 425 permanent traffic data collection devices across the state. Thirty-six sites are weigh-in-motion sites that collect truck axle weight; the remaining sites collect either total volume or volume-by-class.

3.6. Crash-Records Information System (C.R.I.S.)

The Crash-Records Information System (C.R.I.S.) contains detailed information on traffic accidents that have occurred in Texas.[10] C.R.I.S. data can be accessed via the following methods:

- Annual Summary Reports (PDF)
- Automated Crash Data Extract Files (XML)
- An Interactive Query Tool (Figure 3)
- An On-Line Request Form
- The CRASH (Crash Reporting and Analysis for Safer Highways) System

In the Austin District, the City of Austin (CoA) is post-processing raw C.R.I.S. records daily to ensure the accuracy of critical information, such as the latitude and longitude of an incident, for specific incident types. Such data will be shared back with the public through the CoA open data portal.[11]

[10] https://www.txdot.gov/inside-txdot/division/traffic/crash-statistics.html
[11] https://data.austintexas.gov/Transportation-and-Mobility/-UNDER-CONSTRUCTION-Crash-Report-Data/y2wy-tgr5

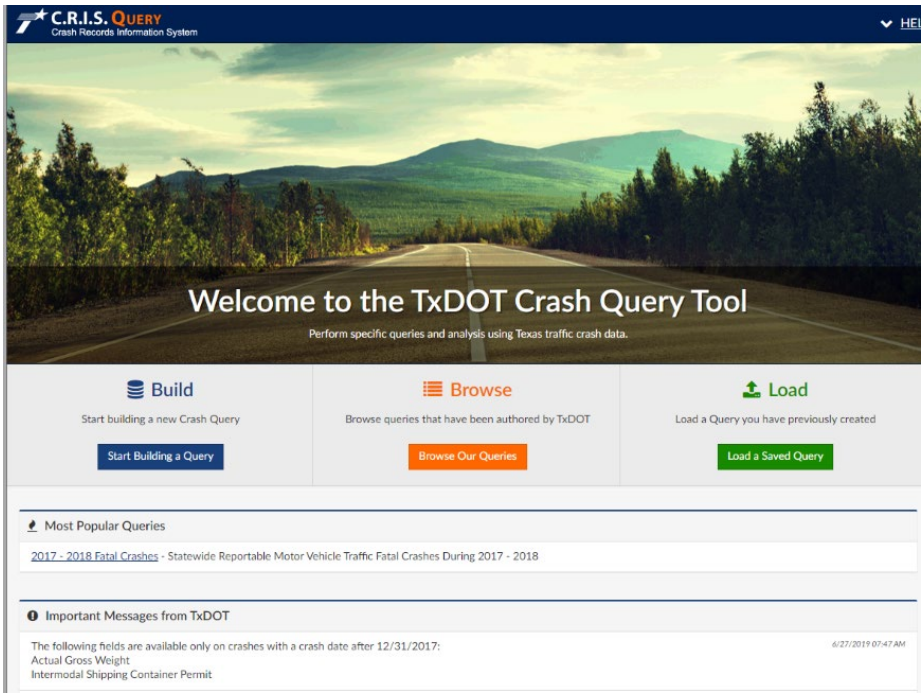


Figure 5. User interface for online C.R.I.S. data query tool.

C.R.I.S. data includes:

- Crash location (county, city, highway, etc.)
- Crash details (drug-related, speed-related, construction-related, etc.)
- Unit details (car, bus, bike, train, etc.)

Additional C.R.I.S. documentation is available through the TxDOT website:

- http://www.txdot.gov/inside-txdot/division/traffic/crash-statistics.html
- https://cris.dot.state.tx.us/
- http://www.txdot.gov/driver/laws/crash-reports.html
- https://cris.dot.state.tx.us/public/Query

3.7. TxDOT Open Data Portal

The TxDOT Open Data Portal (Figure 6) is a web-based ESRI ArcGIS site that provides access to a wide variety of geospatial data.

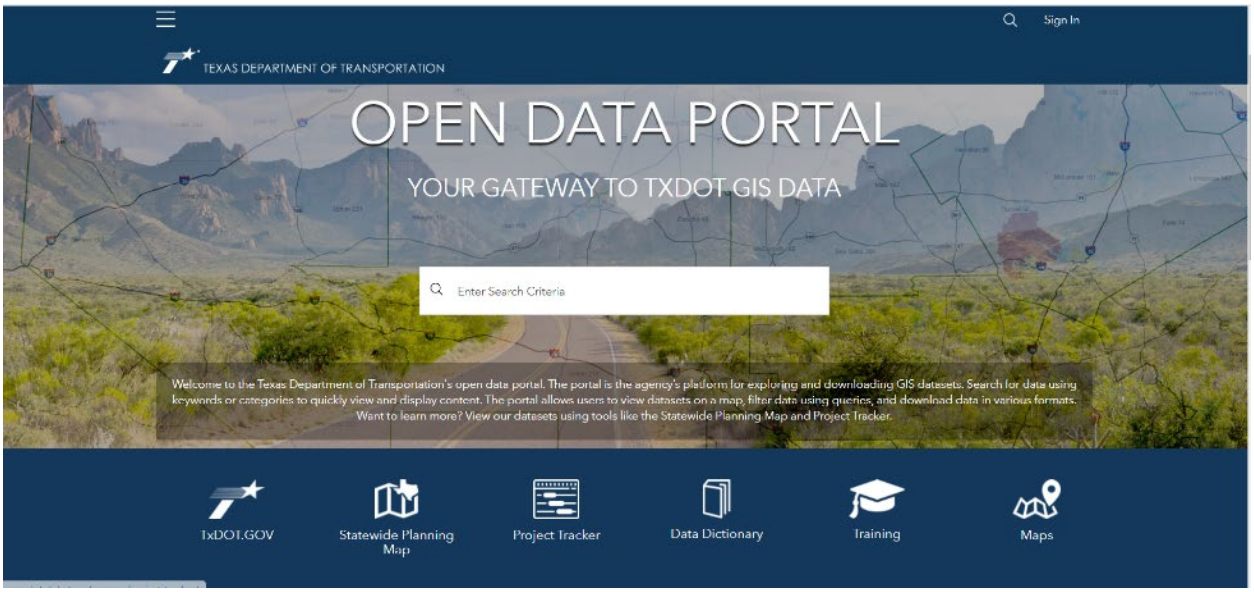


Figure 6. Landing page of TxDOT's Open Data Portal.

The data on the Open Data Portal can be accessed via the interactive web interface or via the ArcGIS web APIs. The latter provide access to data including:

- Traffic count locations and AADT
- Statewide analysis model
- C.R.I.S. Data
- Drive Texas Line Data
- Drive Texas Point Data

The Open Data Portal contains information on categories including traffic, safety, infrastructure, assets, planning, and roadway network and system performance.

3.8. Third-Party Traffic Speeds and Travel Times

A number of vendors offer traffic speed/travel time data based on GPS information from participating vehicles, which is usually processed and, in some cases, combined with other data sources.

3.8.1. Inrix

Inrix provides a suite of data products and tools to support traffic analysis and management.[12] A number of transportation agencies use INRIX travel time estimates within their traffic

[12] https://inrix.com/products/ai-traffic/

operations workflows, including TxDOT. Some INRIX data is currently archived in TxDOT's Data Lake (Section 4).

Inrix data consists of one-minute probe-based speeds, reported for segments of varying length. In this context, probe-based speeds refer to an average speed value for each segment, which is computed by sampling individual vehicle speeds within the segment. Inrix defines the geometry of the segments and includes it in the metadata provided along with the data and updated four times a year. Inrix data includes information that may be used to understand how representative the reported values are, based on sample size and other data characteristics. While the data can potentially cover the entire network, the sample size on some roads is likely to be insufficient to generate reliable estimates.

The Federal Highway Administration makes a subset of Inrix data available at no cost to transportation agencies through its National Performance Metrics Research Data Set.[13] The data is available at five-minute intervals, and it covers the National Highway System only. The data is updated monthly, rather than in real time. Transportation agencies and their partners/consultants may request access to this dataset and download data periodically.

Similar data is available from other vendors, including HERE.[14] Google hosts an API that may be used to extract point-to-point travel time data, and various vendors offer products built based on Google's platform.

3.8.2. Waze for Cities Data

Waze for Cities[15] Data is a platform that provides participating agencies with access to crowd-sourced data, such as the location of incidents and bottlenecks (Figure 7). Cities may use such data in their network planning, management, and operation workflows. Participating agencies are expected to reciprocate data of their own, including street closures and construction information.

[13] https://npmrds.ritis.org/analytics/help/#npmrds
[14] https://www.here.com/products/traffic-solutions
[15] https://www.waze.com/ccp

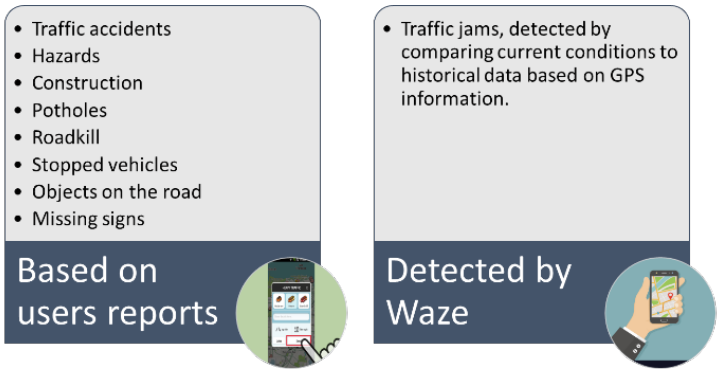


Figure 7. Data shared through the Waze data feed.

Waze shares its data in real time through an API, and it has recently enabled access to historical data through Google Cloud. The new data access option also makes available data analysis tools for participating agencies (Figure 8). The first 10 GB of data storage and one TB of resources for data analysis are free to Waze's "Connected Citizens" program participants, but there will be a fee for additional data storage and analysis needs.

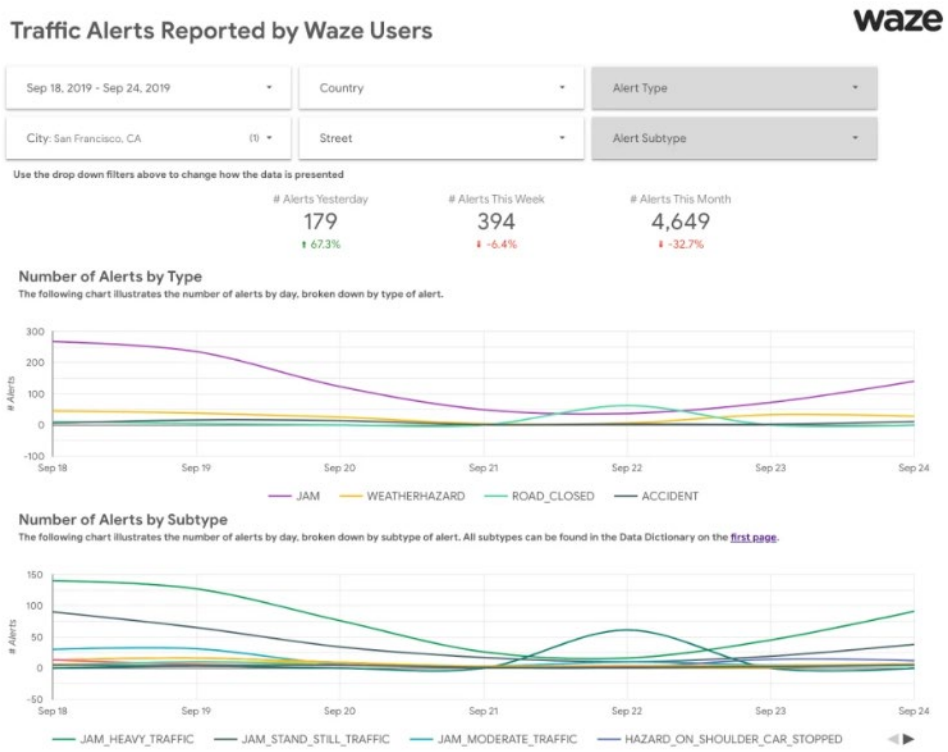


Figure 8. Example of Waze for Cities Data Dashboard.

TxDOT is part of the Connected Citizens Program, which grants access to the Waze for Cities Data. CTR is still investigating current data access/sharing protocols within TxDOT, and between TxDOT and its partners, and will report on it in the updated version of this document.[16]

3.9. Transit Data

In recent years, transit agencies have started using widely accepted data standards to share information concerning their systems, including routes layout, fares and schedules, as well as real-time vehicle location and ridership data. Figure 9 describes transit data types and potential uses in the context of transportation network planning and operations.

While not all agencies make their data publicly available, many medium/large urban areas in the United States share at least basic system information using GTFS. In Austin, Capital Metro provides the following data:

- GTFS: Updated periodically throughout the year
- AVL: Streamed in real time in two formats
- APC: Updated twice a year

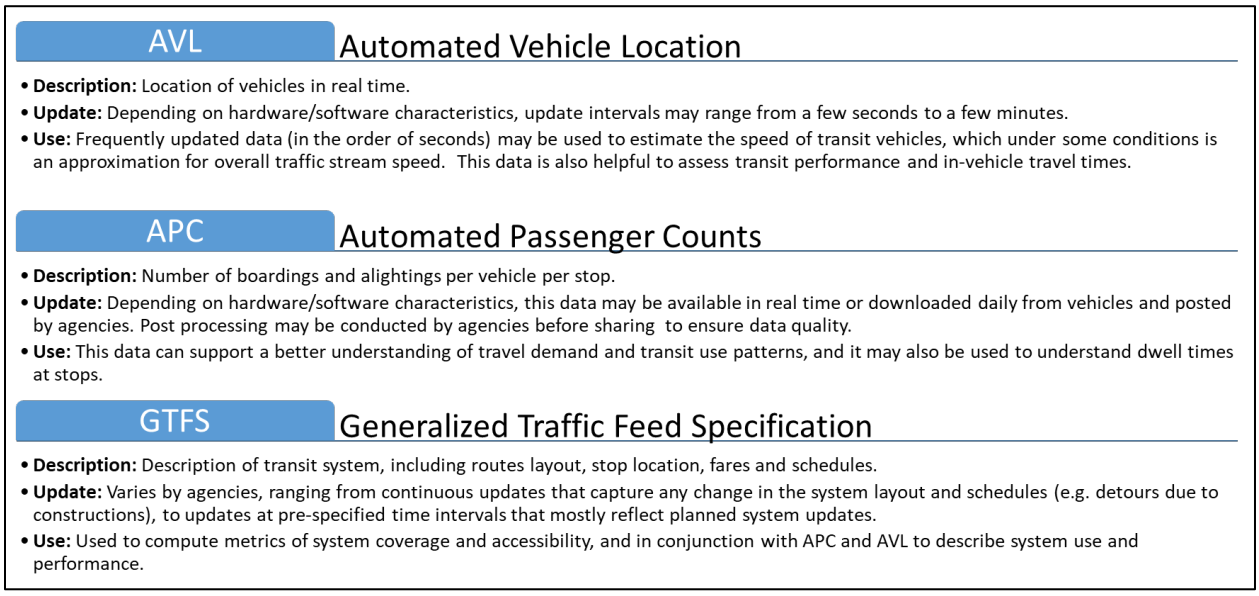


Figure 9. Transit data standards.

[16] Researchers are working with Brent Eastman to better understand TxDOT's access and use of Waze data.

3.10. Experimental/Project-Specific Data with Limited Availability

Aside from the data sources described in previous sections (which are expected to be available to some degree throughout Texas), some districts/divisions may have access to data from ITS devices that are associated with specific projects, or that are being tested in a research project. While such data is not necessarily streamed through Lonestar or even easily accessible, it may be of interest in the context of this effort. Further, the analysis of experimental data sources to support specific use cases may lead to valuable insights when considering promoting their wider adoption and use with TxDOT. Some examples of available data based on ongoing CTR research include:

Austin District smart work zone trailers (SWZT): Trailers devices have been deployed alongside the roadways at more than fifteen locations along I-35 and US 183 through Austin. They capture vehicle volume and speed. While SWZTs are mobile, selected units have been placed at fixed locations for the last 12+ months, allowing for an understanding of recurrent and non-recurrent traffic conditions. The data is streamed through Lonestar and is also being archived to support the analysis of work zone impacts and the planning of future construction work. During important closure events, additional trailers are deployed for 24–48-hour periods to extend the system coverage. Their data is incorporated in the corresponding work zone impact estimation.

Austin District ITS sensors: There are several semi-permanent sensors placed on poles/signs that collect vehicle volumes, speeds, and type by lane every minute.

Weather sensors: TxDOT RTI Project 0-7007, "Weather-Responsive Management Strategies," is evaluating sensor technologies with low-cost weather stations (Abilene District), bridge deck surface characterization (Austin District), on-vehicle roadway surface characterization in both districts, and passive infrared surface temperature and vehicle counting sensors. While these data sources are separate from systems that tie in with Lonestar, a major effort within the project seeks to successfully integrate with Lonestar, and to deploy the emerging online GIS toolset within the TxDOT summary.

3.11. Conclusions

This chapter presents an outline of data sources currently available to TxDOT, which were considered for use in the context of Project 0-7034 to support specific use cases of interest to TxDOT by leveraging artificial intelligence techniques.

This effort is centered on applications that may enhance transportation network operations, and integrated corridor management in particular, and emphasis has been placed on data sources that can describe traffic conditions, safety, and the roadway network. There is additional information that would be beneficial from the perspective of corridor management,

such as the time and location of planned and unplanned roadway closures, and traffic signal timing plans. The performing agency did not find a standardized source for such data at the time of this report.

CTR coordinated a facilitated roundtable discussion among third party data providers in collaboration with TxDOT's Texas Technology Task Force in April 2020. The roundtable explored how emerging transportation data sources can be used for a wide range of applications. Participating data providers included telematics and mapping companies, telecommunications companies, sensor and Internet-of-Things (IoT) companies, vehicle manufacturers, and tier one automobile part suppliers. Each participant was asked to provide an overview of data sources, opportunities for collaboration, and information on use cases. CTR also conducted a survey to provide insights into the desirable characteristics and components of a successful data procurement strategy, reported in Chapter 4.

Chapter 4. Use Case Prioritization

This project aimed to provide a demonstration and understanding of the concrete and tangible benefits that artificial intelligence (AI) may offer to transportation system owners and operators when considering the vast volumes of data currently and prospectively collected. The project focused on understanding methodologies ranging from simply AI algorithms to more advanced machine learning and deep learning methods. This project identifies both traditional and prospective data sources and maps them to meaningful applications that are utilizable to transportation agencies. The research team, along with TxDOT, explored possible applications of AI in transportation and organized them into the following six categories: system and service planning; asset management; system operations; communication and information; business administration; and public safety and enforcement.

Preliminary work focused on developing a broad, high-level summary of the state of the art/practice in AI and its relevance to TxDOT and forming the basis for conducting in-depth analyses of one or two selected applications. The applications selected for this effort are based on the priorities identified by the research team after meeting with the project team and following the discussions held during a workshop organized for this project (P2).

An AI strategy that prioritizes use cases that are directly linked to agency goals is helpful for allocating limited resources, driving toward outcomes that are actionable in advancing the agency mission, and gaining leadership support. The following categories of use cases have been identified as priorities and need further refinement.

- **Traffic Management**: Applications would use collected data to analyze traffic patterns, generate solutions and optimal control policies, and establish rules for applying them to the transportation system.
    - Detecting non-recurring congestion: Data from hard-braking events, adverse weather, crowdsourced data, and queue warnings can be used to detect real-time traffic information and notify traffic management centers of anomalies. With this additional information, TMCs can automatically adjust their signal timing plans along adjacent corridors to alleviate the congestion.
    - Integrated corridor management (ICM): A combination of cameras, vehicle detection systems, and vehicle probe data could be used to improve signal timing along frontage road intersections and nearby arterials for major corridors. A use case with indicated higher interest from TxDOT staff invokes reinforcement learning (RL) techniques to learn optimal traffic signal timings based on observed roadway conditions. By providing multiple examples of simulated data

that reflects conditions encountered over a wide variety of traffic scenarios, an RL agent should learn the traffic signal configurations which maximize the rewards given to the agent. To do so, the RL agent is trained using state data on the roadway's speeds,[17] volumes, and capacities, and rewards based on volume data. The model is developed such that it should provide a straightforward transition to implementation with real-world traffic signal controller meeting the relevant NTCIP standards.

    - Comparative travel times: A combination of historic and real-time vehicle probe data can be used to evaluate travel times along major corridors, enabling TMCs to update their dynamic message signs to publish comparative travel times and enable drivers to make informed routing decisions.

- **Safety and Incident Management**: Applications that can make emergency services and public transportation safer and more efficient.

    - Identifying safety hotspots: Data from crash records, crowd-source platforms, roadway configuration data, weather and event data, and social media can be used to identify areas with relatively high incident rates. Verification of incident reporting data from crowd-source platforms such as Waze could help validate the utilization of this information in public agencies.

    - Detecting incidents and improving clearance times: Additional analysis with data above could identify leading indications for incident occurrence, factors contributing to incidents, and design response strategies such as optimal rerouting.

    - Analyzing work zone impacts: Analyses could help to understand the work zone safety and mobility impacts of their roadway projects. Results could help agencies better design work zones or analyze impacts across multiple work zone and design strategies.

- **System Resiliency and Pandemic Impacts**: Given the timing of current events, it may be of interest to use data and analytical methods to assess the impact on the system from developments such as the COVID19 pandemic or other events, such as Hurricane Harvey, flash flooding, etc.

---

[17] Initially, space-mean speeds are used in training, but spot-mean speeds could be substituted later.

- Travel behavior trends: Data from cameras, probe vehicle data, transit authorities, and more could help to understand household travel behavior trends during the pandemic.
- Impact on freight and supply chain: Roadway sensor data and data from truck fleets could help to understand the impacts on freight operations, including travel times, routing, trip frequency, and origin and destination trends.
- Adverse weather events: Using historical travel and system data, analysis could discover and characterize the impacts to travel due to events such as evacuations, road closures, and emergency response strategies.

Based on the identified priorities and considering current data availability and CTR team expertise, the three use cases were selected for prototype implementation: safety hotspot detection and evaluation of traffic pattern changes due to the Pandemic, real-time traffic signal optimization, and real-time short-term travel time prediction. The three prototype machine learning models implemented by CTR span three of the most popular types of machine learning techniques: supervised, unsupervised, and reinforcement learning. The first two methods are implemented on a connected vehicle (CV) dataset provided by Wejo and focus on understanding the impacts of the pandemic on trip-making patterns and safety, respectively. The third model is tested on a simulation environment and is designed to support the optimization of signal timing plans in response to prevalent traffic conditions, using speed and volume data comparable to that collected by ITS sensors.

Chapter 5. A Strategic Approach to Data Acquisition

Information technology and computing have been advancing rapidly in recent years, translating into opportunities for various sectors. In particular, artificial intelligence (AI) applications can significantly improve upon existing transportation engineering methods. These benefits include improvements to corridor management, understanding traveler behavior, detecting incidents, and distributing information to road users. Rapid technology advancements have made it difficult for public agencies, such as the Texas Department for Transportation (TxDOT), to stay up to date. Among the larger field of AI applications, there is a subset that can serve TxDOT and provide a positive return on investment. The research team has been working collaboratively with TxDOT to identify and prioritize applications based on the organization's needs, capabilities, and priorities.

This process of identifying and prioritizing use cases is one component of a larger process where TxDOT is working to advance their use of data. The Data Acquisition Plan proposed in this chapter may act as a foundation for the development of a formal data strategy. The research team developed a survey because TxDOT personnel are likely to have the best insight on how the organization can progress and where change is most needed. The core goals of the survey were the need to identify high priority applications, to understand internal TxDOT sentiments about current data practices, and to learn about barriers for implementing new methods. The results of the survey have been translated into a number of key insights and recommendations for TxDOT. This is an exploratory effort that provides TxDOT ideas on how to build out a more comprehensive data acquisition strategy to serve the organization.

The chapter is organized into the following sections:

- **Survey Development**: This section provides background on why a survey was needed and how the team went about building it.
- **Survey Results**: This section details the results gathered from survey participants and identifies some key takeaways.
- **Data Acquisition Strategies**: This section translates the insights gained from the survey into actions that TxDOT can take.
- **Conclusion**: This section provides a summary of key findings and identifies next steps.

5.1. Survey Development

The research team's survey creation process has been broken down into design considerations, question refinement, and distribution methods.

5.1.1. Design

The data acquisition survey was created to leverage the existing knowledge base within TxDOT, the personnel who are working with data on a frequent basis. The research team opted for a survey because of the ability to gather information from a larger group of people. TxDOT is a large organization, it is important to get input from people working in various Districts and Divisions to identify unique and/or collective experiences, challenges, and sentiments. One goal of the survey was identifying high priority applications that are implementation ready, so that TxDOT has an idea of projects to pursue that are expected to have a high payoff and grow support for future efforts. In addition, it was important to understand how TxDOT personnel is interacting with and thinking about data. This can be used to identify strengths and limitations in the existing system. From start to finish, the survey design process consisted of question drafting, refinement, survey distribution, and follow-up interviews.

Going into question development, the research team made it a priority to align their objectives with the construction of the survey. The team wanted to gather a variety of information and was able to leverage question structuring to generate valuable results. To gather background data, respondents were asked to rate their data expertise and evaluate TxDOT's relative standing in their use of innovative data methods. These questions were meant to provide high-level information, so a single-choice format was appropriate. Single-choice questions are easy to participants to understand and yield results that are easy to interpret, making them valuable in situations where more detail is not needed. Next, it was important to get an understanding of what applications are the highest priority. In designing this, it was challenging because ranking provides relative measure but is ineffective on a large list of options. The solution was to ask respondents to select three top applications, then use a follow-up question to rank these three applications. This allowed the research team to have a larger initial list of applications and still get insights about the relative priority of the top three applications. To understand sentiments and decision-making influences, multiple-choice questions were more appropriate. Multiple-choice allows for more variation in responses, but still limits responses to a designated list. These questions allowed the research team to identify broad trends without requiring in-depth responses from participants. Throughout the survey, text-entry questions were included to provide an opportunity for respondents to share more details. All of these entries were optional, due to the way that response rate drops off with the inclusion of numerous text-entry questions.

Thoughtful design is key to the success of a survey, as results are only as good as the questions from which they are derived. Survey design goes beyond the questions being asked and considers how the structure of a survey can influence the quality of responses. Questions that were not essential were made optional, with some requesting a response if left blank. In the event of a response request, the survey software would pop-up a notification that the participant has not answered a question before allowing them to advance, but this could be dismissed. The research team understood that surveys are best suited to provide a high-level of understanding that can reveal common sentiments among participants, but they are not well suited to providing detailed insights. The research team kept this in mind during development and focused on questions that provide a broad sense of data at TxDOT. As a supplement to the survey results, participants were able to opt-in for a brief one-on-one interview where they would be able to elaborate on their answers and provide feedback on draft strategies. These discussions allowed time to dig into details that were not captured in the survey and rounded out the research team's understanding.

5.1.2. Refinement

Knowing that the survey should be brief to be effective, the research team developed an initial set of questions before refining to what was distributed. To avoid abandonment and rushed responses, the survey was kept short—containing 10 questions, not including the entry of contact details. The questions around priority applications required particular attention in this process. The information gathered for *TM2.1—Literature Review* was used to generate a list of applications that TxDOT could implement in the near-future. Asking respondents to rank these applications is made difficult by the fact that they are often known by many names and the names may not fully communicate their purpose. To account for this, each application was accompanied by a brief, non-technical description, roughly a sentence long. The refinement process also included the removal of any questions that were redundant or likely to confuse participants. During this process, the research team worked closely with TxDOT's Strategic Planning (STR) Division. Insights from this collaboration helped the research team to strike questions that were unlikely to yield valuable results, to reword questions that were confusing, and to add more options in sentiment questions that could reflect some nuances. The full list of questions asked within the survey is provided in Appendix A.

5.1.3. Distribution

The target audience for this survey is someone within TxDOT who works with data on a frequent basis and/or has some knowledge of the subject. To reach this group, the research team leveraged existing mailing lists within TxDOT. The survey was sent out to the Connected and Autonomous Vehicle (CAV) Workgroup, the Tableau Community of Practice, and others

involved in GIS mapping or road data usage. In addition, the research team identified a handful of knowledgeable individuals, with roles that allow them to implement new technologies, and sent them the survey directly. This process yielded 32 responses and provided enough information for the research team to close the survey.

5.2. Survey Results

The results of the survey provide insights across a few categories. Reporting of result has been broken down into background data on survey performance and participant information, the selection of priority applications, and insights pertaining to participant sentiments and opinions.

5.2.1. Background Data

The survey was completed by 25 TxDOT employees working across a variety of divisions and districts within the organization. Among these respondents, there was representative from ten different divisions and four districts. Divisions represented include Strategic Planning (STR), Traffic Safety (TRF), and Planning (TPP). The districts that participated ranged from rural (Odessa and Brownwood) to urban (Houston and Austin). The diversity of districts and divisions represented translates into a confidence that a variety of perspectives are reflected in these survey results. While the respondents work in a variety of roles, over 85 percent indicated some level of knowledge and experience with regard to data. Figure 10, below, shows the full breakdown of expertise levels concerning TxDOT data reported by respondents. The responses from individuals that indicated a lack of expertise did not significantly differ from the full group, except a higher frequency of selecting lack of skillset and authority as barriers to implementation.

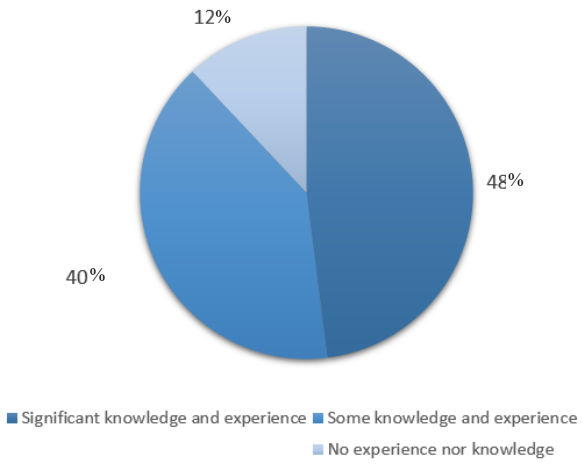


Figure 10. Experience and knowledge levels among respondents.

The research team wanted to know how TxDOT personnel perceives the standing of the organization in comparison to its peer states. Participants were asked to rate Texas as ahead, on par with, or behind other states with regard to innovative data methods. If the participants had frequently rated the state as behind other states, this may reflect a dissatisfaction with TxDOT's data advancements. However, this question did not yield clear results with over 55 percent of respondents saying they are not sure where Texas ranks. Of those who did provide a rating, it was more likely to be ahead of or on par with—indicating a lack of TxDOT employees that think the state is lagging behind in their data related efforts. A better sense of internal sentiment would require further surveying, but it was not critical for the purpose of this survey.

5.2.1.1. Metadata Gathered

While reviewing the results, four responses were removed because they only answered the first question. Three more responses were incomplete, but the answers they did provide are included. The other 25 surveys were fully completed, a promising sign that most survey respondents engaged with the survey. All respondents that fully completed the survey provided a TxDOT email address which allowed the team to verify their role at the organization. During the design phase, the research team estimated that the survey would take 5–10 minutes to complete thoughtfully. The available metadata suggest that most durations lined up with this. However, many values were likely an inaccurate reflection of time spent taking the survey. Qualtrics, the survey platform used, measures duration as the time between starting the survey and finishing the survey without accounting for any breaks. Many participants had duration times that suggest they do not reflect time spent focused on the survey—a handful of times upward of one to two hours were reported.

5.2.2. Priority Applications

One goal of this survey was to understand what applications TxDOT prioritizes most highly so they could feed into pilot efforts and/or implementation research. The list is meant to encompass applications that are relevant to TxDOT and could be deployed in the next one to three years. Looking to the results from applications that were included in the list generated by the research team, there are a several levels of priority that can be identified. Figure 11, below, shows the breakdown of responses with values that can range from 0 to 28. Figure 12 shows the breakdown of how the top three applications were ranked relative to each other. The applications were broken down into three tiers of priority level (high, middle, low), so the graphs in Figures 11 and 12 have been sorted and color-coded to reflect these tiers. Breaking the results down by these tiers:

- Identification of Safety Hotspots and Integrated Corridor Management are the high priority applications among this list and, therefore, could be strong candidates for implementation.
    - They were selected as a top three application the most often.
    - Both were more likely to be ranked first priority, rather than second or third.
- Applications of mid-level priority were Incident Detection, Asset Management, Travel Trends & Forecasting, and Disaster Response.
    - These applications were selected less often and less likely to be ranked first priority.
- The lowest priority were Flood Detection and Truck Parking availability—likely a reflection of the situational importance of these applications.
    - Respondents from the Houston region selected Flood Detection more often.
    - When selected, these applications were ranked second or third.

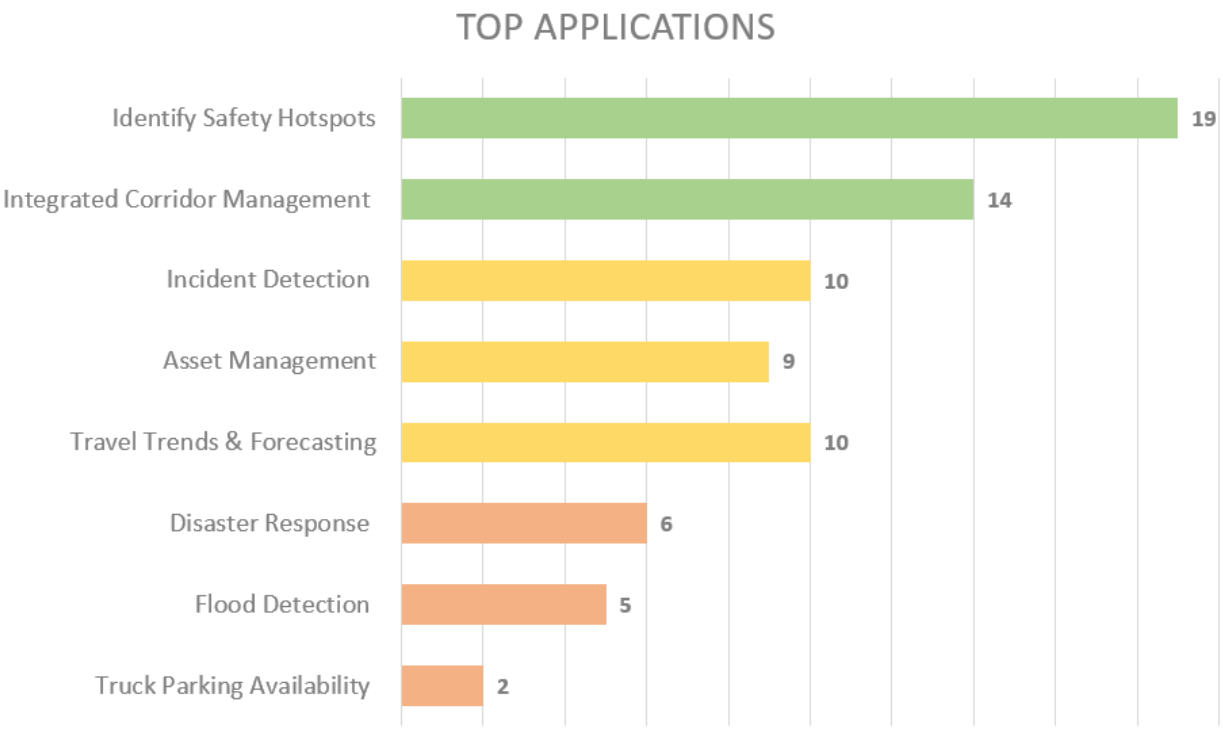


Figure 11: Priority application choices.

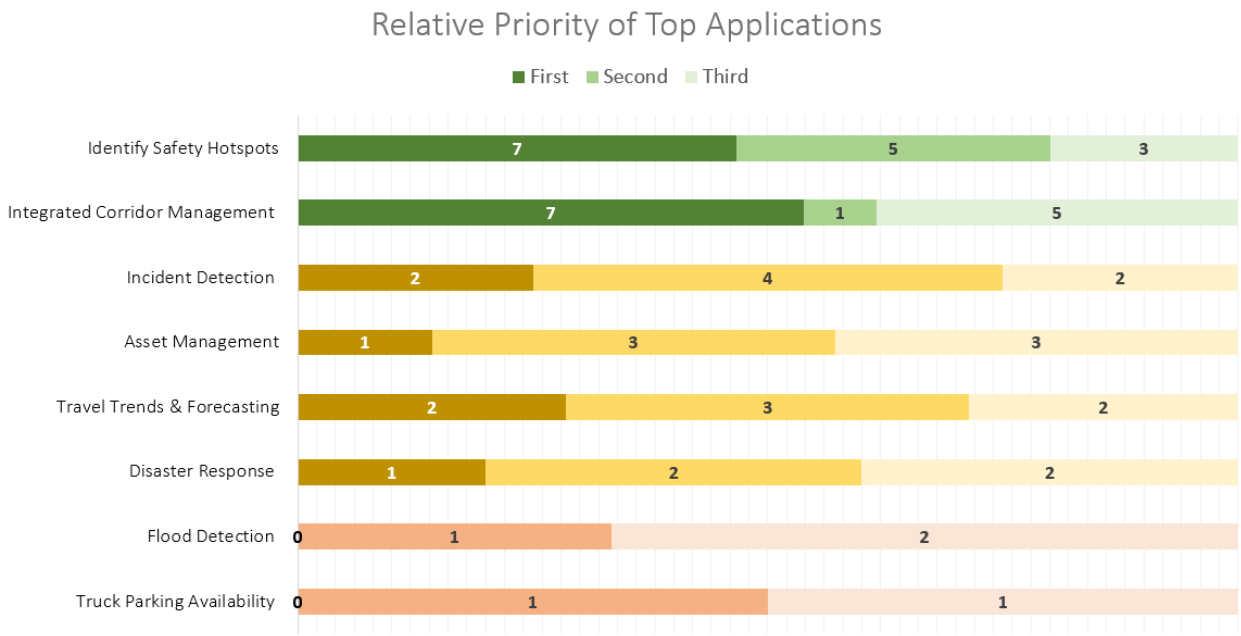


Figure 12: Relative priority rankings of top applications (darker shade indicates a higher ranking).

The list of applications in the survey was not comprehensive, so the survey included an opportunity to identify other applications. Respondent suggestions included predictive travel time, project management tracking, and resource optimization tools. In a follow-up discussion, the research team was able to talk to the respondent who suggested the project management tool. They elaborated on this concept as a tool for generating estimates to provide to contractors during the project letting process, leveraging historic data of project performance to provide an idea of how this project will fare. The research team is not aware of tools that currently target this challenge, but the respondent stated that centralization of data and development of data standards could be a big help. In terms of predictive travel time, this is a promising application that could be implemented by TxDOT to improve the information provided on dynamic messaging signs (DMS) during periods of congestion. The resource optimization tool was a suggestion for a scenario-planning application that would allow someone to allocate resources (funding, staffing, etc.) across a number of scenarios and predict the outcomes. It would be a great tool for asset management and maintenance planning efforts.

5.2.3. Respondent Sentiments

The survey contained a set of questions that were aimed at learning about underlying perceptions. These questions provide insight into what challenges are occurring or, at the very least, are perceived to be occurring. Also, they provide a look into what employees value—information that is helpful for members of the private sector who want to highlight features of their work that matter to the public sector. Since these results are generated from a list of

options, they should be not thought of as comprehensive and there are likely factors that were not revealed here. The results that were gathered do show trends that are helpful in developing programs and strategies for the future.

In the first sentiment question, respondents were asked to identify up to three factors that played a large role in their decision of what applications to prioritize and their relative ranking. The results, shown in Figure 13, reveal a high number of participants selecting the impact and alignment factors. This can be translated into a preference for applications that will create a noticeable difference from traditional methods and for applications that are aligned with existing priorities. Equally interesting, is the fact that funding was one of least selected factors. Funding is often seen as a significant barrier to change, but this reveals there is still a stronger preference for applications that can make a difference, regardless of any difficulty relating to funding. The stronger preference for readiness and sustainability versus novelty can be interpreted as support for applications that are shown to yield consistently strong returns, instead of an attraction toward novel concepts or techniques that are less proven. In sum, these selections reflect a desire to invest in applications that will provide strong results and can be sustained over time. Applications will need to be thoroughly tested and show promising results before there will be broad acceptance of incorporating new tools or techniques.

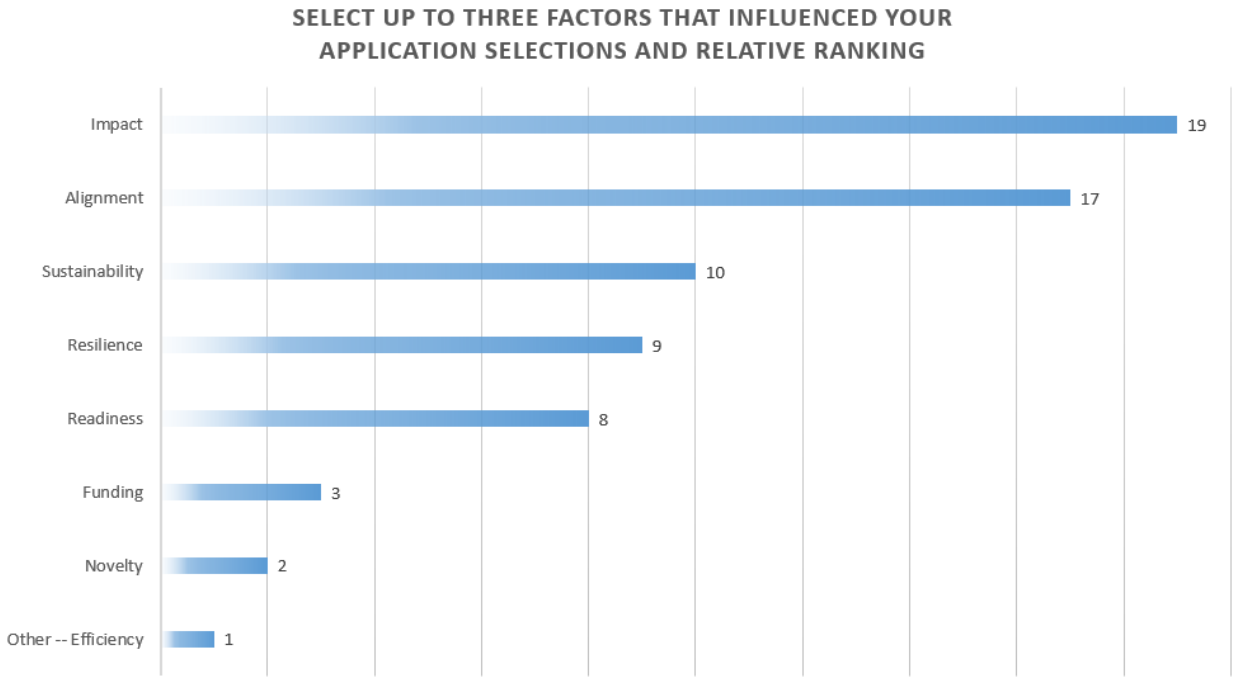


Figure 13: Factors that influenced top application selection and ranking.

Next, respondents were asked to identify up to two barriers they had experienced in working to improve TxDOT's data tools or management. Efforts that they have engaged with previously did

not need to be connected to AI. The breakdown of barriers identified by respondents is shown in Figure 14, the most common barriers were a lack of skillset and a lack of awareness. The barriers of skillset and awareness reflect a trend that can be seen across transportation agencies and many other fields. Technology is advancing quickly, making it time consuming to keep up with advancements. Having the awareness that data analysis tools and techniques are advancing is not the same as having the ability to implement changes. Some respondents indicated a challenge with understaffing, a factor that further restricts the ability to explore new methods. The other common barriers, limited funding and lack of authority, add further complications to these scenarios. In the realm of funding, it can be hard to justify these efforts when competing with requests for funding required to maintain the costs of current methods. Lack of authority is a common challenge for an organization the size of TxDOT, it is often hard for employees to know who they should talk to for help. During a follow-up conversation, a participant elaborated on the challenge of being 'bounced' between various contacts and losing time from the lack of clearly delineated roles. These challenges do not reflect poorly on TxDOT; they instead illustrate the level of difficulty that surrounds the process of modernizing a large organization.

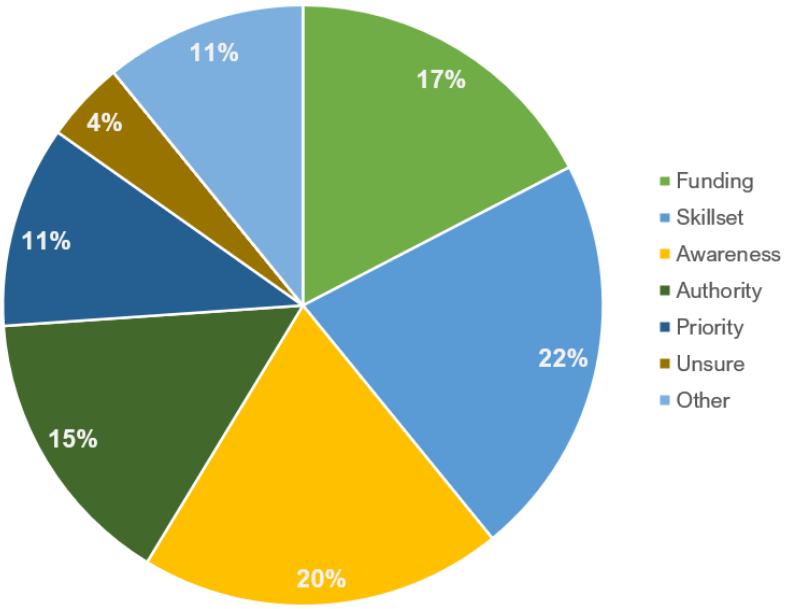


Figure 14: Results for barriers to implementation.

The last quantitative question in this subsection pertained to the evaluation of new tools and techniques. Respondents were asked to consider a hypothetical situation wherein they were analyzing a private sector solution and identify up to three things that would factor heavily in their decision. The most common selections, by a significant margin, were evidence and

visualizations. These selections reflect a high value placed on proven solutions that provide a user-friendly interface. The preference for visualizations is well-aligned with the fact that TxDOT employs a lot of people that are not, and do not need to become, data experts. Visualizations quickly translate data into easy-to-interpret formats that can be shared and discussed. While tables are compact and easy to construct, they require significant mental energy to translate into concise insights. This difference is magnified as the quantity of data grows. The support for evidence reflects a preference for tools that can be trusted to provide a return on investment and, possibly, consideration that evidence can be used to support a request for funding. There is also a notable interest in endorsements, pilots, and evaluations. Pilots often offer an opportunity to work through any integration complications and provide a chance to work with the solution before committing to a longer contract. Endorsements and evaluations come from trusted sources, providing a sense of confidence. Interestingly, there was limited value placed on the maturity of an application. This survey is unable to reveal if maturity is valued more generally, but it was not a significant factor in these results. Similar to the responses from the earlier question about decision-making factors, cost was not identified as a priority. Considering the insights on barriers to implementation, it can be said that the cost of new solutions is not a driver in identifying what should be implemented but, instead, is a barrier to putting that solution into practice.

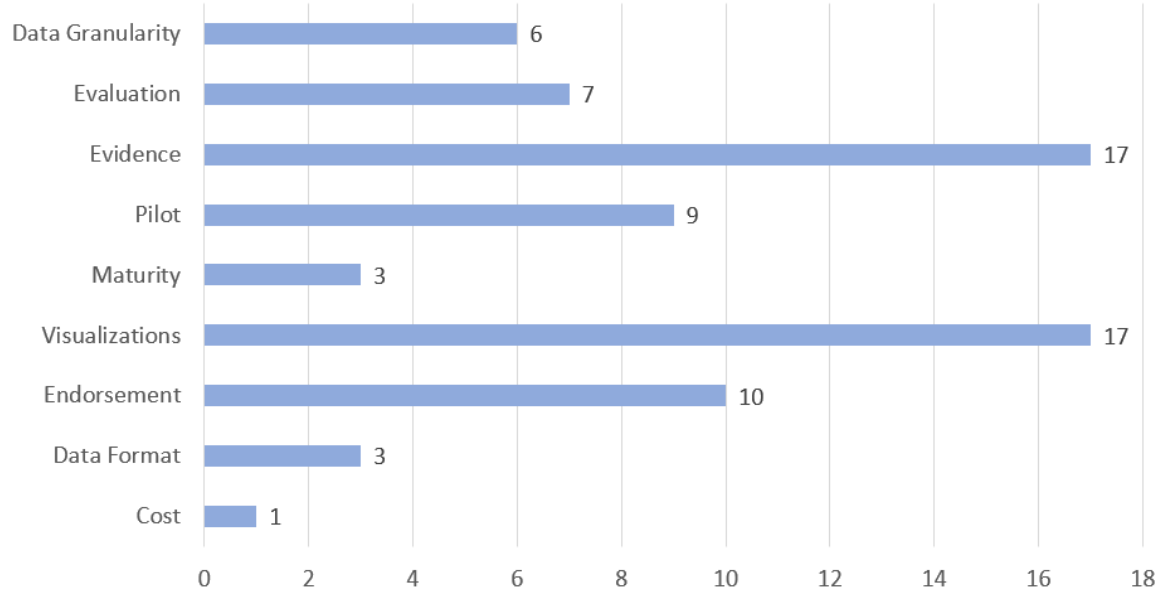


Figure 15: Factors when evaluating new data source or analysis tool.

At the end of the survey, before opting in or out of a follow-up, participants were offered a chance to provide some general input into putting together a data acquisition strategy. The full, unedited responses are available in Appendix A, but here they may be paraphrased or combined to highlight key insights. A common refrain, reiterated in the follow-up interviews, is

a need for centralization of data and governance of data. Without a central hub and standardized data formats, much time and energy are lost to converting or creating data sets. There may be more use cases that TxDOT can support, but it is nearly impossible to identify them without bringing together the existing data—both that being stored and the streams actively generating data. Another key message was the importance of holistic strategies that target high impact use cases and implement them in a timely manner. Implementing solutions that are well-aligned with existing priorities and high impact can be low-hanging fruit—providing high returns with little investment. The time to develop and deploy a solution should be considered, as a project that takes too long may result in a loss of momentum. In terms of personnel management, there were requests for better access to training and coordination of communication between divisions and districts. Employees often feel they are left to navigate with little guidance, so it is important that TxDOT provide continued support across the organization that enables collaboration and growth.

5.3. Data Acquisition Strategies

The survey results provided a set of valuable insights, next these insights need to be translated into actionable strategies. The research team's recommendations have been broken down into three categories: (1) assets, (2) personnel, and (3) policy and planning.

5.3.1. Digital Infrastructure and Data Assets

The success of innovative data analysis applications is built upon effective data storage and up to date hardware. TxDOT is generating more data than ever, with this volume of data expected to continue growing in the coming years. To keep up, TxDOT will need to build out their data storage capacities to some extent. The primary options include:

- **Localized hardware: TxDOT can invest in their own servers for data storage**. This approach would allow greater control over data management, particularly important in the realm of data security. The drawback being a higher upfront investment and continued cost of maintenance.
- **Cloud storage: TxDOT can outsource much of their data storage needs.** There are numerous cloud storage options, TxDOT already uses SharePoint for some of their needs. This requires less investment, but it involves security risk and increased latency to access data.

Likely, the ideal solution for TxDOT will be a combination of the above. Local storage can be used for data that needs higher security and low latency, with other needs served by cloud storage options.

For TxDOT to leverage their data assets, they need an Enterprise Information Management strategy in place. Data needs to be easy to access and work with, as friction in this process results in wasted resources spent integrating data sources. Personnel in various divisions and districts would benefit from access to TxDOT's data inventory. The success of this concept will also require TxDOT to implement more data governance:

- **Centralized Repository:** TxDOT has begun efforts to develop a data warehouse that will improve storage of and access to data. The central storage solution will need to support queries that allow personnel to easily access samples for analysis. Districts and Divisions can maintain smaller storage of recent, granular data that is not used by others. For frequently accessed data, the hub can reduce redundant storage and make data easier to find. Improved storage and access will allow TxDOT to better leverage their data.
- **Standardized Formats:** Data streams need to have defined formatting to ensure consistency across efforts. TxDOT has already done great work in furthering this with their role in developing the Work Zone Data Exchange (WZDx). This mindset can be applied across data sources and improve the ability to analyze data across time, Divisions, etc. It may not be feasible to standardize all of TxDOT's data immediately, so they may need to work with personnel to identify low-hanging fruit (data that can easily be standardized and will provide noticeable improvement to operations).

5.3.2. Personnel

In the survey, there was a high percentage of respondents who reported a lack of authority to make changes. The distributed power structure of TxDOT enables it to operate efficiently, but it can make it harder to navigate. This difficulty manifests internally and externally:

- **Internally**: Individuals struggle to identify the appropriate individual when trying to interact with other divisions and/or districts.
  - *Challenge*: In a follow-up interview, a participant recounted a time where personnel within a district was trying to collect data by hand using an ArcGIS map. The division that manages this map was able to write a query and pull the needed data within minutes, but the individual assigned the task was not aware. This story illustrates how individuals can spend time performing tasks inefficiently when intra-agency contact is not well established.
  - *Solution*: TxDOT can create a directory that identify point(s) of contact for various needs—i.e., manager of an internal tool. To develop an effective tool, personnel across multiple districts and divisions should be consulted. This resource will need to be easy to access and updated regularly.
- **Externally**: Many private sector companies that want to work with TxDOT have difficulty identifying the appropriate contact within a division or district.

- *Challenge*: Companies do not know the best point of contact within the organization. This results in two ways: (1) emails are never sent, TxDOT may miss out on a valuable opportunity or (2) emails are sent to whomever they can reach, resulting in many emails to those who are not responsible for these efforts.
- *Solution*: Divisions and districts that interact with the private sector on a regular basis (i.e., TRF and STR) can collate a list of contacts. This can be made available on the TxDOT website and would help to streamline communication.

The survey results showed a high proportion of respondents feel a lack of relevant skills presents a barrier to implementing new data analysis tools and techniques. TxDOT can address this by providing training and/or an information repository that makes knowledge more accessible. In survey responses and during a follow-up interview, participants requested that training be designed thoughtfully and to fit the needs of those who need it. Training may cover a range of topics from more introductory information (What is AI? How can it be used in TxDOT?) to practical use of tools and techniques in practice at TxDOT (What are the capabilities of INRIX? When should I use Tableau?). Training should be designed to suit the needs of personnel. This may come in the form of modularity, an ability to select what training is applicable; flexibility, allow training materials to be accessed at any time; and variety, present the information across multiple formats. TxDOT may not be able to eliminate the skillset limitations through training, but it is possible to reduce the gap.

Respondents reported a similar issue in lack of awareness surrounding new technologies and solutions available. It does not make sense for all TxDOT personnel to spend large quantities of time learning about each new technology or technique, yet without knowledge of these new opportunities they cannot be pursued. TxDOT can build upon their existing success with communities of practice. In a follow-up interview, the participant said they really enjoy the community of practice they have joined for Tableau and would like to see TxDOT expand this concept to encourage more conversation around data. TxDOT could create a community of practice or otherwise facilitate conversation on the topic of new data analysis tools and innovative solutions. Creating this forum would allow people to come together and leverage their collective knowledge. By collaborating, individuals often come away with ideas they would not have come up with on their own.

5.3.3. Policy and Planning

The large size of TxDOT can make implementing new solutions a slow process, but this pace needs to accelerate and there must be room for innovation. The research team recommends that TxDOT develop a streamlined pilot and/or evaluation process for new data sources and analysis tools. Pilot projects provide an opportunity to explore the value of a new solution to

TxDOT that is difficult to measure otherwise. Recently, a group within TxDOT worked to develop the INRIX contract that has provided easy access to an immense pool of data. The INRIX contract should be seen as a valuable precedent, as it made access ubiquitous across the organization and extended to their public partners. TxDOT can use this as a base to develop a more streamlined process, allowing this model to extend beyond the INRIX contract. These pilots can focus on high impact investments that can show results within a small timeframe, solutions that show promise can then be evaluated for a larger implementation where it makes sense to do so. By streamlining the process, TxDOT can trial more new technologies instead of spending time investigating many different options without being able to conclusively understand how it would operate. The pilot strategy limits the risk taken on by TxDOT, the short length of the contract means that costs can be kept low and there are limited loses from any effort that does not succeed. TxDOT can engineer this process to focus on proven solutions that have demonstrated some promise and applicability, aligning with the priorities identified in the survey. It could translate into rapid-paced improvements that inspire further changes.

5.4. Conclusion

The results of this survey and insights from follow-up interviews show that TxDOT has made great strides in using data and pursuing innovative solutions. This progress can be built upon to further progress and help the organization transition into a holistic data strategy. TxDOT will need to expand upon their efforts to formalize data standards, efforts to bring personnel together, and efforts to create contracts that enable innovation. The key takeaways for TxDOT can be summarized as:

- **Pipeline Process**: **Create a procedure where priority applications can be identified and explored in a pilot project**. Building upon progress from the INRIX contract, TxDOT can create a streamlined method for scaling new solutions. The priority applications from this survey, safety hotspot identification and integrated corridor management, could be a jumping off point for this concept.

  This process should include development of evaluation criteria for private sector data sources and analysis tools. Standardized criteria will improve efficiency and fairness.

- **Internal Collaboration**. **Build on existing successes, create a new working group that focuses on data and implementing new solutions.** Working groups and other facilitated conversations enable individuals to come together and share their collective knowledge. Make it easier to communicate across divisions and districts by clearly designating authority and point(s) of contact.

- **Data Policy**: **Establish clear data governance policies to centralize data management and standardize data formats**. The array of data generated and managed by TxDOT must be centralized to enable ease of use and efficient analysis. Well governed data makes it easy to perform internal analysis and generate insights. Establishing standards early on, such as in the WZDx project, allows TxDOT to set a standard for their peers.

Chapter 6. Data Collection for Prototypes

This document describes the data that CTR will use in the practical applications to be conducted for this project. The following sections present the general characteristics of each dataset, corresponding spatial and temporal coverage, and how the researchers will access, process, and stage the data. The final datasets will be put together after the prototype model development phase is completed, at which point CTR may update this technical memorandum with information specific to each use case. Alternatively, such information may be provided in the technical memorandum describing the model validation and refinement for each use case.

6.1. Data from Smart Work-Zone Trailers

The Austin District uses smart work-zone trailers (SWZT) to collect speed and volume traffic data during lane closures related to work zones. These devices collect one data point per minute, which indicates the spot-speed (in miles per hour) and the volume at the location of the sensor. The data is streamed in real time by two different providers, and then broadcasted through Lonestar. In collaboration with the Austin District's GEC, CTR has developed a data pipeline to store all streamed data in a relational database.

Figure 16 presents the location of 33 SWZTs for which data is available starting on November 4, 2018, plus permanent ITS sensors. Table 7 presents the data tables used to maintain the SWZT data.

Table 7. SWZT Data Tables

Table	Field	Comments
sensor_location	sensor_name	
	lat	
	long	
	Net_ID	Internal identification, not used.
Sensor_archive *Stores last 72 hours of data*	Volume	
	Sensor_name	
	Net_ID	Internal identification, not used.
	XML_Header	For debugging, not used.
	Occupancy	
	Speed	
	Direction	
	Agg_type	For internal reference. Set to average for all rc
	Archive_time	Timestamp including date and time.
Sensor_archive_history	Same fields as sensor archive but storing data older than 72 hours.	

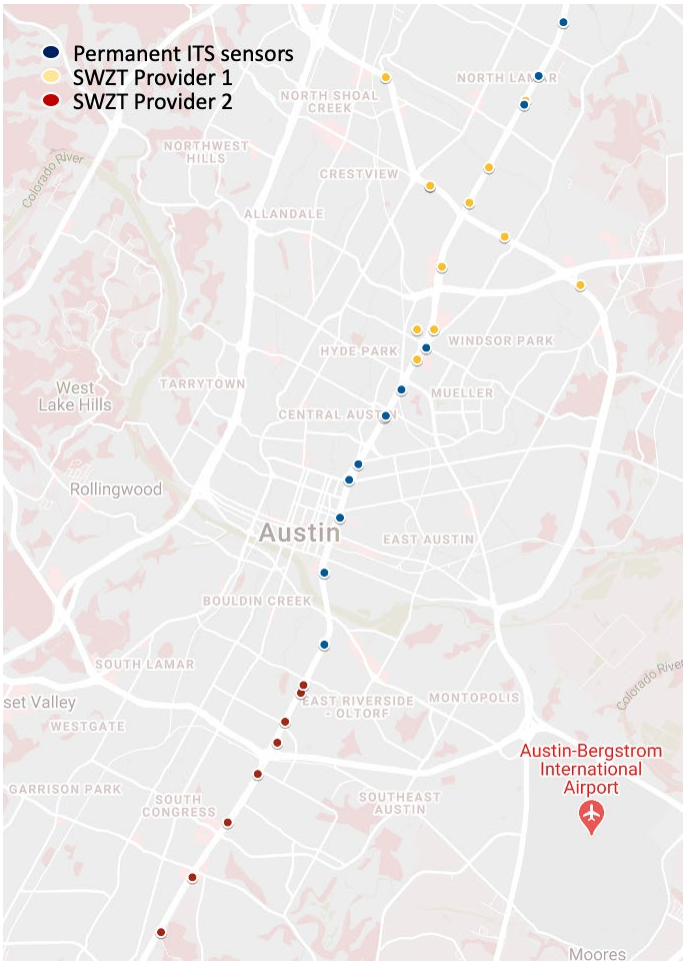


Figure 16. Location of SWZTs and selected permanent ITS sensors in the Austin District.

CTR has analyzed the collected data, and identified some issues that must be considered when using this data:

Missing data: Each sensor should produce one data point per minute; however, there are occasionally hours of the day for which the sum of received data points is less than 60. Although on average most sensors collect 90 percent or more of the data they are expected to, data points may be missing from the data sample used in our applications. Missing data is typically the result of sensor malfunction, and it may affect the calculation of total throughput and bias our estimates of average speeds. Appropriate methods will be used to minimize the impact of any missing data, such as imputing missing volumes based on neighboring data.

Repeated records: A single data point is expected for every minute in which sensors are active. However, when a momentary loss of communications between the sensors and Lonestar occurs, several data points may be broadcasted simultaneously, all of which are assigned to the same minute. While the data in each of these points is valid, the timestamp is incorrect, which may lead to small errors when aggregating the data for the analysis of a specific time interval.

Repeated records represent less than 1 percent of the data collected in each sensor up to October 2010. While these errors are not expected to be significant, repeated records will be corrected in the data sample used in this project.

Constant speed values: We have observed that occasionally the reported speed values do not change for several consecutive minutes, which is likely to reflect a sensor error. For this project we will try to avoid including days where the standard deviation of the reported speed values is zero for one or more hours.

6.1.1. Data Sampling and Processing

Data from SWZTs will be used to test and validate the reinforcement learning application for optimizing traffic signals. This data may also support the development of supervised learning methods to understand the impact of freeway closures on arterial streets, and to predict the impact of freeway closures. CTR may also consider testing classification models to identify abnormal traffic conditions.

CTR has received samples of incident reports from CTECC and is expecting to select a subset of such incidents for the development and testing of the algorithms. Based on the dates and times of the selected incidents, CTR will select a subset of records in the existing database and create files or a project-specific database for the purpose of testing and validation.

Some of the considered applications may require further data cleaning, and the research team will describe such methods in TM 4 (or amend this document).

6.2. Probe-based Speed Data

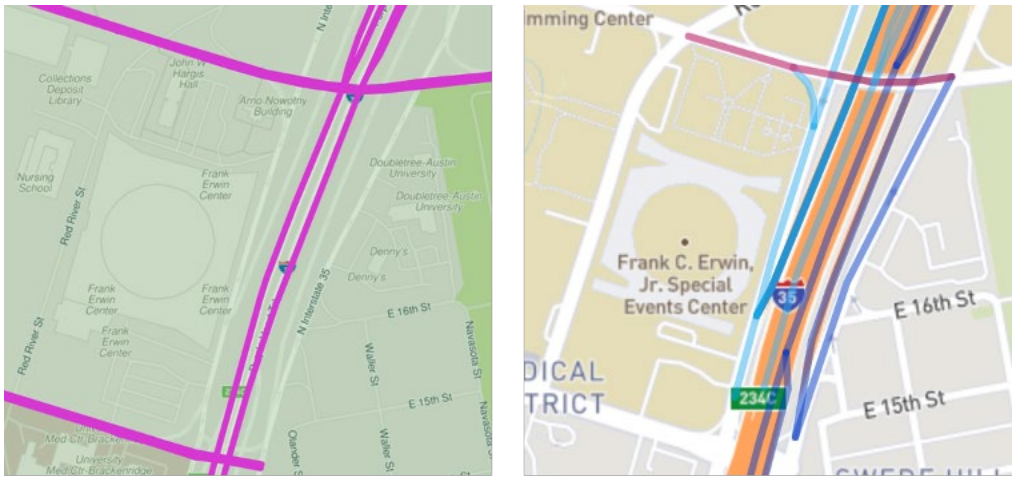
CTR has access to probe-based speed data from two sources: the National Performance Management Research Data Set (NPMRDS) and INRIX.

Probe-based speed data consists of average speed values on pre-defined roadway segments, obtained by analyzing the position and speed of in-vehicle devices for a set of participating vehicles. The data is often streamed in real time and updated every minute.

Both datasets used in this project are collected by INRIX; the NPMRDS data is made available for free to transportation agencies and their consultants by the FHWA through RITIS (ritis.org). The NPMRDS data is not provided in real time, but rather updated monthly. The data is aggregated to five-minute intervals and mapped to segments denoted TMCs (Traffic Message Channels), which are an industry standard to convey location information to communicate traffic information.

The NPMRDS data cover only roadways in the National Highway System. INRIX data obtained through the provider is streamed in real time every minute, and it covers most of the roadway system in urban areas (Figure 17). Further, the data is reported over segments defined by the provider, called SD segments, which are typically smaller than TMCs.

Probe vehicle data has some limitations: the measured speeds do not correspond to any particular vehicle, and they may not reflect the conditions throughout the segment (e.g., a segment may be congested on one end and free flowing on the opposite end). Nevertheless, the extensive spatial and temporal coverage and ease of access makes this type of data a valuable tool to support network management and decision-making.



a) NPMRDS b) Inrix XD segments

Figure 17. Example of availability of probe-based speed data along I-35 when accessed through the NPMRDS (a) and when purchased from the provider (b).

6.2.1. Data Sampling and Processing

Probe-based data will be used to test and validate the reinforcement learning application for optimizing traffic signals. This data will also support the development of supervised learning methods to understand the impact of freeway closures on arterial streets, and to predict the impact of freeway closures. CTR may also consider testing classification models to identify abnormal traffic conditions.

For this project, CTR has gained access to NPMRDS data, and also received permission from the City of Austin to use a data sample of INRIX data on the arterial system.

Data samples will be downloaded through the web interface based on the availability of incident/lane closure data. We expect to download at least one month of data from both datasets, spanning the entire city of Austin, and use 15-minute-level data. The data will be staged in a database, and it is not likely to require additional processing or cleaning, since the

provider performs quality analysis and control. Table 8 describes the tables to be used to store INRIX data.

Table 8. INRIX Data Tables

Table	Field	Comments
segment _geom	Seg_ID	
	O_lat	Latitude of start point.
	O_long	Longitude of start point.
	D_lat	Latitude of end point.
	D_long	Longitude of end point.
	Seg_dir	Direction of travel.
	Street_name	
	length_mi	Segment length in miles
	seg_geometry	Geometry of straight line between start and end visualization
segment _data	Seg_id	
	Date_time	Timestamp
	Speed_mph	
	Ttime_min	

6.3. Data from Permanent ITS Sensors

The PM has access to sample data from permanent ITS sensors placed on TxDOT's infrastructure. These sensors collect vehicle speeds and traffic counts by lane, and also perform vehicle classification. CTR has identified 12 sensors on I-35, which cover a section of the freeway not monitored by SWZTs (Figure 17).

ITS sensor data is affected by similar limitations to those described in Section 2, although the volume and speed data collected by these sensors is expected to be more reliable than the data from SWZTs due to the sensing technology.

6.3.1. Data Sampling and Processing

ITS sensor data will be used to complement SWZT data if needed. CTR has explored the availability of data using a one-month sample and the data seems promising. There's currently not a pipeline in place to collect this data continuously, and CTR will request samples based on the need of specific applications.

6.4. High Frequency Connected Vehicle Data | Basic Safety Message

CTR has negotiated access to a dataset collected by Wejo, a United-Kingdom-based company with a US presence. Wejo has agreements with US automobile companies to share and mine high-frequency connected-vehicle data for insight on travel demand, behavior, system operations, and more. The shared data is similar to the basic safety message (BSM) standard developed by SAE J2735 BSM. BSM has two parts. Part 1 contains the core data elements, including vehicle size, position, speed, heading, acceleration, and brake system status; this data is to be transmitted approximately 10 times per second. BSM part 2 contains a variable set of data elements drawn from many optional data elements (availability by vehicle model varies) and is transmitted less frequently.

The Wejo data represents one in every 28 vehicles in the US. Its coverage is approximately 95 percent of US roads. It is transmitted every one to three seconds from the vehicles, and 95 percent of it is sent to customers in less than 32 seconds. Finally, the data is accurate within a three-meter radius. Examples of Wejo data collected include:

- latitude
- longitude
- speed
- direction of heading
- ignition status
- timestamp
- state code
- country code
- vehicle make and model
- acceleration

6.4.1. Data Sampling and Processing

CTR received two weeks of data for the Austin area: one week before social distancing orders, and one after.

The data was be received through a safe FTP transfer and staged in a database to facilitate data manipulation. The database schema, along with any necessary data aggregation and processing methods is described in TM 4, or in a revision to this document.

Potential applications and use cases fall into two categories that are described below.

- **COVID-19 pandemic travel patterns,** including an analysis of travel behavior, traffic volumes, and trends in the weeks leading up to social distance orders and compared to patterns after social distance orders. More specifically, there is interest in analyzing travel speeds on various roadway types, as traffic incidents and their severity have increased since social-distance orders were issued. Additionally, there is interest in understanding trip purpose trends (e.g., recreational, work, essential service, etc.).
- **Connected vehicle data for corridor management,** including exploration of data for applications such as incident detection, advanced detection and prediction of roadway interruptions, ramp metering, queue detection, lane management (managed lanes), wrong-way driving, and more. An area of interest may include estimation of minimum connected-vehicle penetration for accurate parameter estimates (speed, volume, etc.) so that other sensors can be phased or complemented. Other areas of interest may include development of methods to predict future travel conditions based on real-time information, dynamic signal timing, developing traffic smoothing strategies, and intersection safety analysis.

6.5. Additional Data on Austin's Arterial System

CTR is familiar with a number of datasets curated by the City of Austin that may be used to support some of the proposed use cases. These include turning movement counts at intersections collected by Gridsmart cameras, traffic speeds and volumes collected by Wavetronix sensors, and travel time data that may be obtained by analyzing Bluetooth sensor readings.

6.5.1. Data Sampling and Processing

CTR has developed the workflows used by the City of Austin to publish the above datasets in its open data portal and will use the established pipeline to download adequate samples as required by specific use cases. Corresponding data structure, as well as processing and cleaning needs, will be described in TM 4 if these data sources are used in the project.

Chapter 7. Safety Analysis Using Unsupervised Learning

This chapter describes the first of three prototypes developed by CTR based on the use cases identified in collaboration with TxDOT. The first prototype represents an application of a clustering analysis (an unsupervised learning algorithm) to identify safety hotspots along the I-35 corridor in Austin, Texas. The analysis uses high-frequency CV data from Wejo, roadway geometry information, and crash reports from the Crash Records Information System (CRIS) database to develop a roadway typology and compare characteristics before and after stay-at-home orders were issued in March 2020 due to the COVID-19 pandemic. The cluster analysis proved useful, identifying five major clusters in March and April 2020 each, which demonstrated the value of AI techniques to characterize roadways based on different driving behaviors and safety risks. The following describes the data, methodology, and key findings from the cluster analysis.

7.1. Background

Safety remains a significant priority at TxDOT, and new data sources as well as data science methods are becoming available to enable the agency to more readily identify its challenges and allocate its resources accordingly. Current safety analysis relies upon crash reports inventoried by the CRIS database. This information, however, is historic and typically used to conduct analyses once a crash has occurred. TxDOT is seeking opportunities to approach safety more proactively.

Furthermore, the COVID-19 pandemic brought forth new challenges for transportation agencies. Stay-at-home orders were issued on March 24, 2020, and as a result, traffic volumes dropped significantly. Unfortunately, speeding increased and traffic fatalities even increased in some cities. With such a significant disruption to typical travel patterns, it became important to study how different traffic conditions were related to different sets of safety challenges.

CVs are an emerging technology capable of generating high-frequency data that provides insights into driver behavior, trip patterns, and traffic conditions. The data typically includes basic information on the time and position of the vehicle, along with its speed, heading, braking, and acceleration, and additionally other optional information such as windshield wiper status. While the potential value for this data is clear, much remains to be learned about data management, skills and training requirements, and its efficacy for different use cases.

The prototype selected for development represents an opportunity to demonstrate an application of an unsupervised learning algorithm while assessing the value of CV data.

7.2. Research Question

Based on input from TxDOT, the research team focused its analysis on identifying safety hotspots before and after COVID-19 stay-at-home orders were issued. Using a CV data sample, the research team devised a methodology to classify roadways based on different characteristics—driving behaviors, roadway geometry, and crash information—and identify significant safety hotspots. As a proof of concept, the research team chose a limited geographic area and time period for the basis of the prototype.

7.2.1. Geography

The analysis efforts focus on Interstate Highway 35 (I-35) main lanes and service roads from Cesar Chavez to US 290. This scope covers the dense urban environment near the river, but also extends up to a less congested portion of I-35. This allows for some range in driver behaviors, which bears out in the results. Along this stretch, 113 road segments are used for analysis from the Street Centerline file. The cluster analysis could be extended further along I-35 or to other highways but this was not feasible within the time and resource constraints of the project.

7.2.2. Time Period

The analysis was temporally restricted to the evening rush period on weekdays. The rush period was designated as 4:00 to 6:00 PM with data pulled from Monday to Friday in the first full week of March (March 2–6, 2020) and April (April 6–10, 2020). Restricting to this period allowed for the expectation of similar traffic flows across the analysis period. Additionally, limiting the scope of analysis made it feasible given project constraints. Other time periods of interest may be the morning peak period, event traffic (SXSW or ACL), or late evening (higher fatal crash rate).

7.3. Data Collection & Management

This project utilized data provided by Wejo, a connected car data company, to assess the value of artificial intelligence (AI) techniques for analyzing large datasets. A sample dataset was provided for use in this project, but the full catalog of data owned by Wejo is much larger, spanning a broader physical scope and temporal scope. For the purpose of this report, references to Wejo data are tied to the two-week subset of data within the Austin region provided for analysis.

7.3.1. Wejo Data

Wejo data is generated by connected cars that are equipped during the manufacturing process, instead of using aftermarket devices. Data collection is triggered by starting the vehicle and

ended by turning off the vehicle. All data collected provides no direct information about the operator of the vehicle or any passengers.

Wejo maintains two datasets: (1) Movement, which contains records of each ping from a connected car while in operation and (2) Event, which tracks identified behaviors according to specified metrics. The data is generated by the same set of vehicles and one could theoretically derive the Event dataset from the Movement dataset by hand.

7.3.1.1. Movement Data

The Movement dataset consists of pings sent approximately every three seconds while the vehicle is turned on (Figure 18). While this project used historical data, Wejo also has real-time data with a latency of roughly 30–60 seconds. The time between pings and latency of transmissions means this data does not meet the requirements of CV data, but in non-safety critical applications it can provide comparable value. Each ping contains information on time, speed, heading, and location of the vehicle, which is called the basic safety message (BSM).

In addition, data is connected by a journey ID from the time a car is turned on to when it is turned off. Multiple journeys by the same car are not connected to each other. Finally, each ping has information on the make, model, and year of the car in operation. A sample data point is shown in Figure 19.

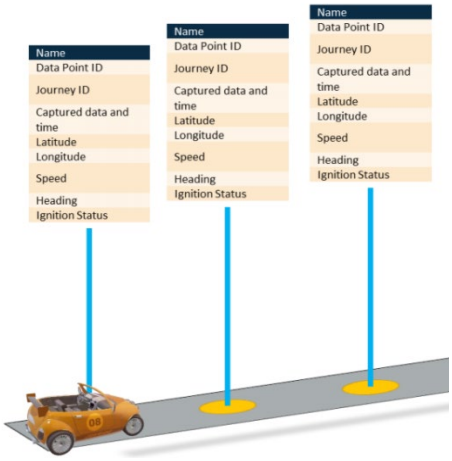


Figure 18. Movement Data Diagram.

Since Movement pings are sent every three seconds, the dataset is massive: approximately 80 GB for two weeks of data. This fine-grained data can be used for in-depth analysis of travel behavior and is particularly valuable for real-time analysis of road conditions (i.e., congestion detection). However, the large volume of data can be cumbersome to work with across longer spans during historical analysis. In these scenarios, the Event data collected by Wejo can be particularly valuable.

{"metrics": {"speed": 52.99, "heading": 207},

"vehicle": {"make": "CHEVROLET", "year": 2018, "model": "SILVERADO", "status": { "ignitionStatus": "MID_JOURNEY"},

"squishVin": "3GCPCRECJG"},

"location": {"geohash": "9v67zs", "latitude": 30.213064, "longitude": -97.754138, "postalCode": "78704", "regionCode": "TX", "countryCode": "US"},

"journeyId": "0e8a87904f0f96d129b0a275afa11fc3b3cd0d17",

"dataPointId": "007feba5-0060-4f19-83a9-f9f566f58c91",

"capturedTimestamp": "2020-04-06T19:25:22.000-0500"}

Figure 19. Example of movement data point.

7.3.1.1. Event Data

The Event dataset contains pings sent when a trigger behavior occurs. In the Wejo data, triggers include hard braking, hard acceleration, speeding (travelling at more than 80 mph), seatbelt status (latched/unlatched), and ignition status (on/off). Each data point contains information on the type of trigger event occurring, the BSM for the moment when the event occurred, location data, and timestamp (Figure 20). This dataset is available only in historic batches and not broadcast in real time.

This project makes use of the Event data because, at roughly 4 GB, it was manageable within the timeframe for analysis and better suited to historic analysis. The work done in this project used the ignition status in one case study and the hard braking, hard acceleration, and speeding events in another case study. For this project, seatbelt status was not used, but in combination with more demographic information it could be used as an unsafe driver indicator.

{"event": {"eventType": "ACCELERATION_CHANGE", "eventMetadata": {"accelerationType": "HARD_ACCELERATION"}},

"status": {},

"metrics": {"speed": 21.13, "heading": 298},

"deviceId": "183212967",

"location": {"geohash": "9v6t1x", "latitude": 30.451826, "longitude": -97.662331, "postalCode": "78728", "regionCode": "TX", "countryCode": "US" },

"journeyId": "T08DF27271583174860428_5",

"dataPointId": "cf744079-4ba2-4d3e-8072-ddf024886073",

"capturedTimestamp": 1583171640000,

"capturedTimestampTz": -21600000}

Figure.20: Example Event Data Point.

7.3.1.2. Data Management

Data was received from Wejo and a database was staged on a server for easy remote access. Event data was staged in a single table while Movement data was split by month to make it more manageable to work with. Data was accessed using PgAdmin where SQL queries were used to navigate the database. Subsets of the data could easily be downloaded and manipulated to allow for exploratory analysis with a limited sample. The numeric format of the timestamp in the Event dataset was much easier to work with than the text string format used in the Movement dataset. Once data was refined down to temporal scope of the case study, it was downloaded as a .csv file to be uploaded to ArcGIS for mapping and spatial analysis.

7.3.2. Roadway Geometry Data

To perform spatial analysis, the project team used the GIS shapefiles made available by the City of Austin (CoA). In particular, the team used the street centerline file to align the Wejo data to roadway corridors. The CoA data was used because it included information about arterial streets which were used in this analysis. The shapefile contained information on the road type, speed limit, and length of the segment used in the case study. This data is easily accessible to TxDOT, which the team considered an important factor in the event that the agency wishes to recreate or expand the efforts of the case study.

7.3.3. CRIS Data

The analysis also incorporated data from the Crash Records Information System (CRIS). This database is managed by TxDOT and includes information on historic crashes from 2010 to present. In particular, the analysis used information on the total number of historic crashes, crashes that occurred during the time period of analysis, crash severity, and contributing factors.

The CRIS data for the City of Austin limits was mapped to the street centerline file and aggregated for each roadway segment used in the case study. Three metrics were obtained for each roadway segment: 1) total number of historic crashes, 2) crashes that occurred from March 2 to 6, 2020, and 3) crashes that occurred from April 6 to 10, 2020.

Next, the team analyzed the crash severity along each roadway segment in the case study. Crash reports typically indicate the crash severity of an incident according to the scale in Table 9, where Crash Severity Code 1 represents "Fatality" and Crash Severity Code 5 represents "No Injury."

Table 9. Crash Severity Codes

Crash Severity Codes	
1	Fatality
2	Incapacitating Injury
3	Non-Incapacitating Injury
4	Possible Injury
5	Not Injured
0	Unknown

Finally, the team analyzed contributing factors for each roadway segment. Crash reports contain fields for officers to report up to two contributing factors. New codes have been added over the course of the historic time period—in particular, more detailed codes 7X-7Y, which describe cell/mobile device usage in greater detail. It should also be noted that contributing factors are not always reported. Since this information is still valuable for safety analysis, the team grouped similar contributing factors together and identified the top six major contributing factors, as shown in Table 10.

Table 10. Top Six Contributing Factors

Top Six Contributing Factors		
ID	**Contributing Factor Code**	**Name**
A	19-20, 72, 74-77	Distracted driving
B	22, 60-61	Speeding
C	44	Followed too closely
D	45, 67-68	Driving under the influence (drug and alcohol)
E	4	Changed lane when unsafe
F	41	Faulty evasive action

7.4. Analysis

7.4.1. Spatial Analysis

To complete a spatial analysis, the project team used ArcGIS Online. The Wejo data within the defined scope was pulled from the database and saved to a .csv file that was then uploaded. Since each data point contains latitude and longitude, it was simple to map the events. In addition, the Street Centerline file from the City of Austin data portal was uploaded to the same map. Once the data was all added to the map, the project team generated heat maps to identify spatial trends and changes between March and April.

The project team created two forms of heat maps, with one using 50’ x 50’ squares to aggregate the data, and the other using road segments buffered by 25’ to either side of the centerline. The grid heat map allowed for analysis across a broader region, but they were not conducive to cluster analysis. This is because many grid squares contained more than one road with different speed limits, number of lanes, and other inconsistencies. The road segments ensured that analysis was constrained to an individual road with consistent geometry. The buffer of 25’ was chosen because the portion of I-35 analyzed never has more than four lanes in either direction, so the 25’ buffer was able to capture the activity on each direction of I-35 without overlapping onto the service roads or the other direction of travel.

7.4.2. Cluster Analysis

7.4.2.1. Normalization

Prior to conducting the cluster analysis, the team normalized the data to compare metrics of different scales. In particular, roadway segments have highly variable lengths; some are longer and thus inherently see more events. To account for this variation, the team normalized hard braking, hard acceleration, speeding, and crash events on a per-mile basis—that is, the number of events was divided by the segment length.

To normalize the crash severity and contributing factors, the team was interested in which characteristics dominated each roadway segment. Therefore, for each roadway segment the team calculated crash severity and contributing factors as a percentage of the total for each roadway segment. This enabled the team to, for example, readily identify segments that contained a high percentage of fatal crashes and recognize segments where distracted driving factored into a high percentage of crashes.

7.4.2.2. Outliers & Data Selection

The team identified two segments that were considered outliers. To carry out the analysis and identify robust clusters, the following two segments were removed from the analysis:

Segment #30558. Only two crashes occurred on this segment historically. Both were Crash Severity Code 2 and one was due to Contributing Factor E (changed lane when unsafe). Consequently, the segment was unique and would form its own unique cluster.

Segment #35995. There were 18,617 speeding events per mile in March on this particular roadway segment. Since this metric was significantly greater than (more than six times) other segments, it would also form its own unique cluster.

7.4.2.3. Cluster Analysis

For the cluster analysis, the team chose the hierarchical clustering method. Hierarchical clustering is an unsupervised learning algorithm that groups similar objects based on their characteristics and can fall into two categories: 1) agglomerative (build-up) methods and 2) divisive (split-down) methods. The agglomerative method was selected for this cluster analysis and follows this procedure:

- Each object is considered to be its own cluster.
- The two objects that are closest to one another are joined to form a cluster.
- The next two closest objects (individual objects or clusters) are joined together to form a new cluster or attach an object to an existing cluster.
- Return to Step 3 until all objects are clustered.

Closeness may be defined in many ways. For this analysis, Ward's procedure (1963) was used to minimize the loss of information associated with grouping individual objects into clusters. The loss of information is calculated by summing the squared deviations of every object from the mean of the cluster to which it is assigned. Ward's procedure assigns clusters in an order that minimizes the error sum of squares (ESS) from among all possible assignments, where ESS is defined as:

$$ESS = \sum_{j=1}^{k}\left(\sum_{i=1}^{n_j} X_{ij}^2 - \frac{1}{n_j}\left(\sum_{i=1}^{n_j} X_{ij}\right)^2\right),$$

where X_{ij} is the value of the object for the i[th] individual in the j[th] cluster; k is the number of clusters at each stage; and n_j is the number of individuals in the j[th] cluster.

To perform the cluster analysis, the team used an Excel add-in developed by DecisionPro, Inc. The software enables an analyst to input data, perform a cluster analysis, and readily interpret the results. The software also produces a dendrogram, which depicts the hierarchical relationship between clusters.

For the case study, Table 11 shows a sample of the data used for the cluster analysis in March. The segments are considered the objects to be clustered, and the data—driving behavior, roadway geometry, and safety information—are considered the characteristics.

Table 11: Sample Table of Data

ID	Wejo Event Data			Roadway Characteristics		
Segment	Hard Braking (per Mile)	Hard Acceleration (per mile)	Speeding (per mile)	Speed Limit	No. of Intersections	No. of Lanes
3312	291.53	69.41	0	50	1	2
5717	851.03	602.82	0	50	1	4
7777	547.53	8.93	0	65	0	4
…	…	…	…	…	…	…

ID	CRIS											
Segment	Historic Crashes (per mile)	CS 1	CS 2	CS 3	CS 4	CS 5	CF A	CF B	CF C	CF D	CF E	CF F
3312	1193.88	0.02	0.14	0.24	0	0.59	0.02	0.01	0	0.01	0	0
5717	567.36	0.00	0.22	0.44	0	0.33	0.06	0.00	0	0.06	0.06	0
7777	1032.58	0.04	0.22	0.20	0.01	0.54	0.10	0.01	0.02	0.02	0.01	0.01
…	…	…	…	…	…	…	…	…	…	…	…	…

7.5. Results & Key Findings

7.5.1. Comparison of Before and After Stay-at-Home Order

On March 24, 2020, the City of Austin issued a "Stay Home – Work Safe" order that closed all non-essential businesses and shifted the vast majority of Austin residents to working from home. Since the Wejo sample pulls from three weeks before and two weeks after this order, it provides a look into how the order impacted travel patterns. In the following months, traffic has returned to volumes that are close to levels observed before the pandemic, but the results of the data analysis confirm that a significant shift occurred in travel behavior as a result of the order. For example, more than 650,000 trips took place in March, but just over 300,000 trips occurred in April. The frequency of hard braking and hard acceleration dropped in correspondence with this trend, but the occurrence of speeding became much more common in April. The project team expects the rise in speeding to be a result of reduced congestion providing the opportunity for drivers to speed on the interstate corridor.

The project team generated a series of grid heat maps to visualize the frequency of each event across the downtown area in March and April. The grid stretches from around 1st Street to 15th Street with I-35 running down the middle. Beginning with hard acceleration, events occurred more frequently in March, with 2,044 events across the analysis period. April had less than half the number of acceleration events, with a total of 888 events. Figure 21 and Figure 22 show heat maps of the hard acceleration events in March and April, respectively, with the scale for each reflecting the reduction in activity in April. The April hard acceleration events are also more concentrated than those in March, which have a more diffuse pattern.

Figure 21: Heat Map of March Hard Acceleration Events.

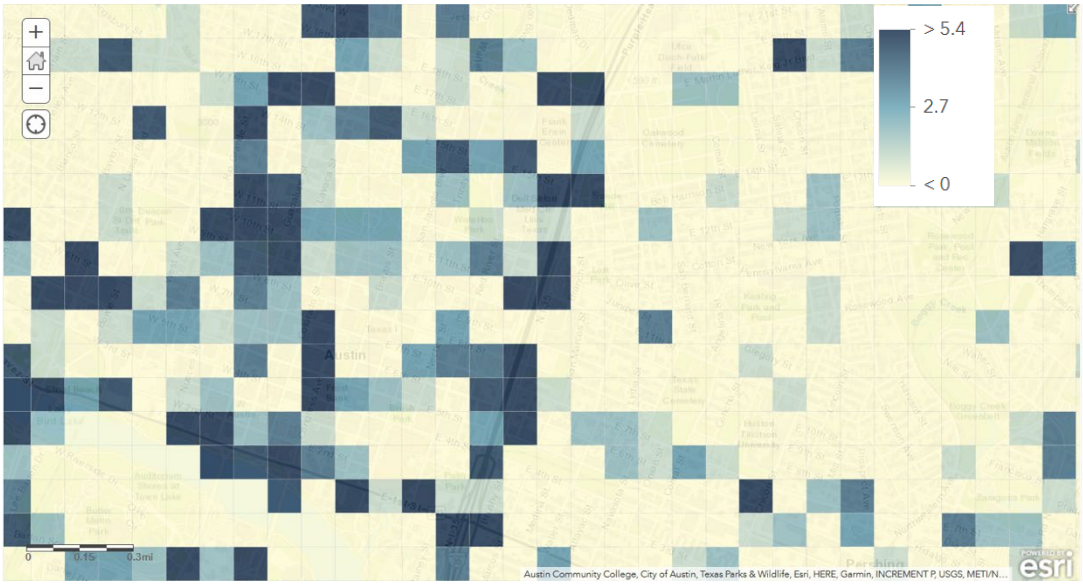


Figure 22: Heat Map of April Hard Acceleration Events.

Hard braking was the most common event analyzed, with almost 25,000 events shown in the heat map area during March. Hard braking declined sharply in April; just under 6,000 events occurred in the same area during April. During March, hard braking was common in the downtown area where congestion is particularly bad during peak periods (Figure 23). In April, hard braking was less common downtown, and a higher portion of the hard braking events took place on the interstate (Figure 24). Since congestion was much lower in April, the reduction in hard braking downtown could be interpreted as easier driving conditions without crowded roadways. In contrast, the increase in hard braking on the interstate could be tied to the rise in

speeding and a need to slow down for other vehicles or to exit the interstate. Since events are not connected to a unique vehicle, it is not possible to link the behaviors together.

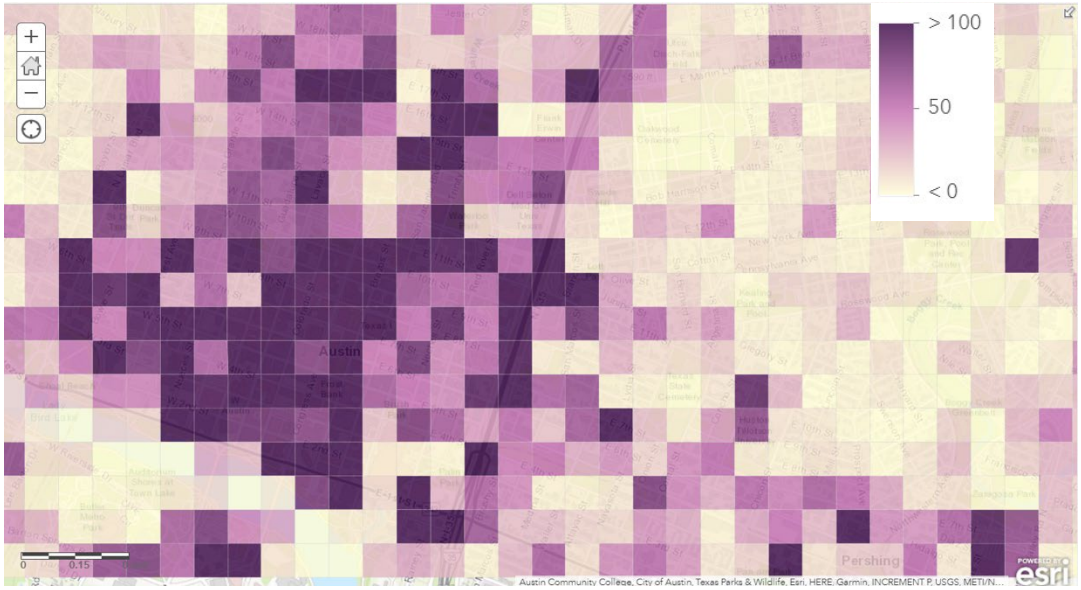


Figure 23: Heat Map of March Hard Braking Events.

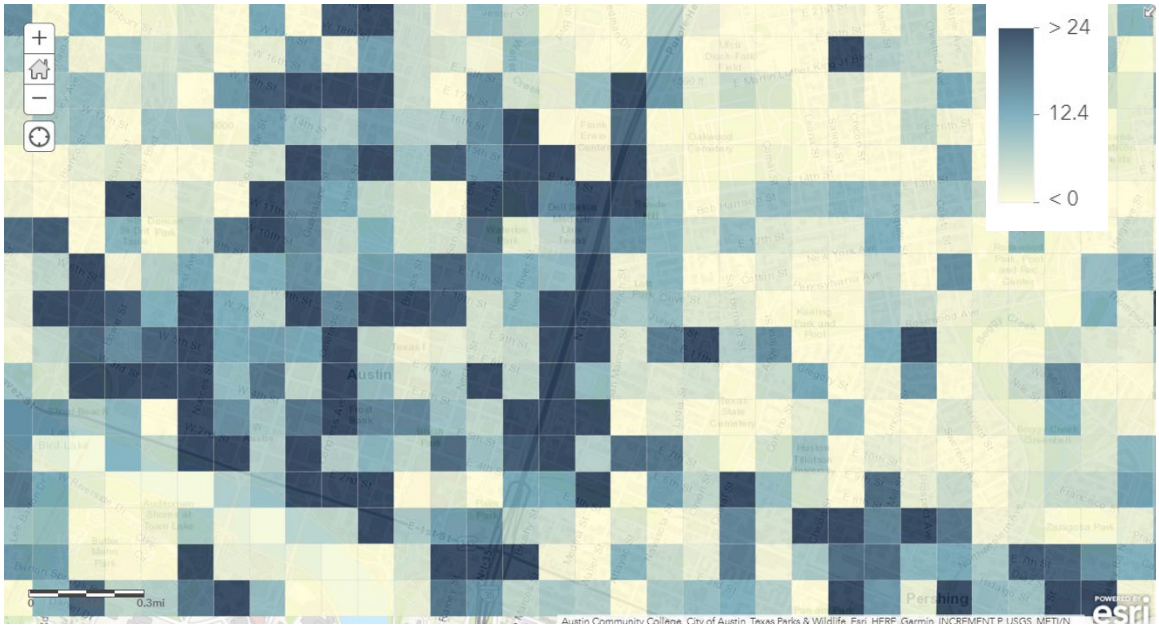


Figure 24: Heat Map of April Hard Braking Events.

The speeding heat map was generated for the same grid area as other events, but the trigger of 80 mph meant that events only occurred on the interstate (Figure 25). In March, there was only one location with any speeding events. However, April saw congestion levels fall and speeding events increase, with 233 speeding events total.

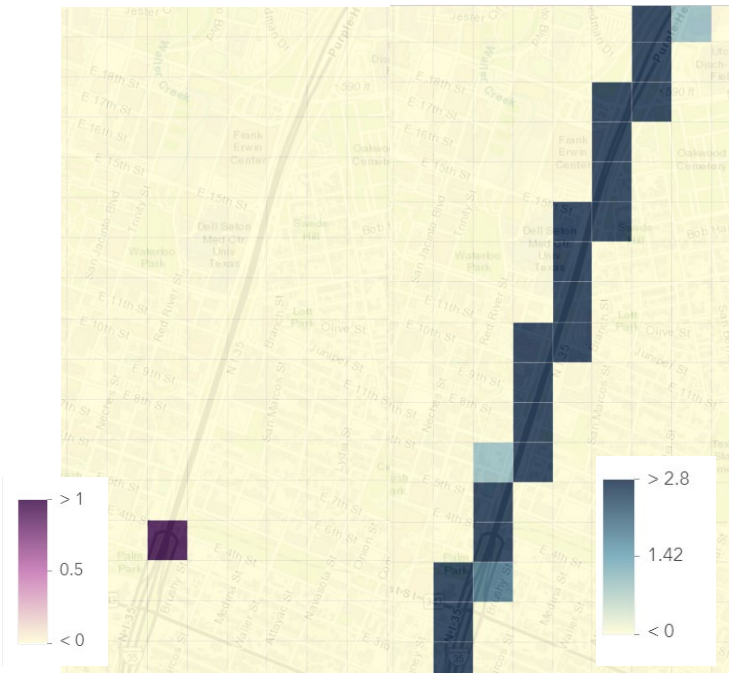


Figure 25: Heat Map of Speeding Event (L: March, R: April).

Crash data pulled from the CRIS database was also mapped to identify any changes between March and April. Collisions were not common in either month, with 20 crashes in March and only three crashes in April (Figure 26 and Figure 27). Since crashes were so uncommon during the period of analysis, the project team decided to supplement with historic crash data along I-35.

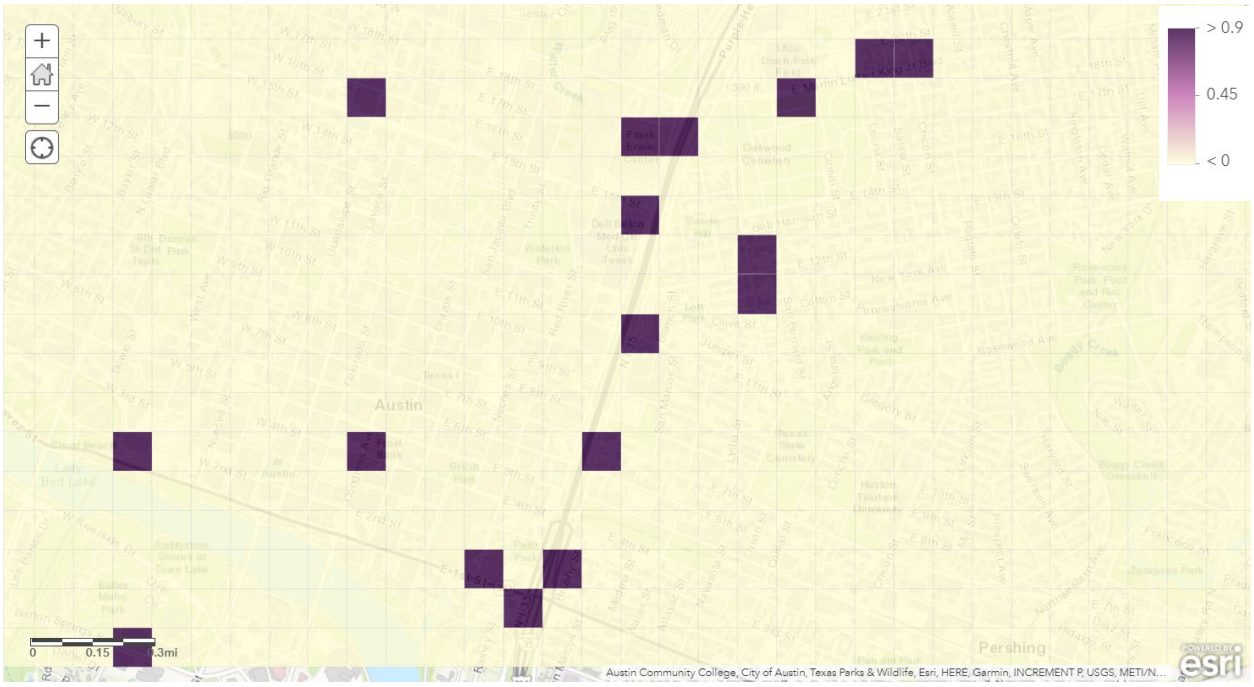


Figure 26: Heat Map of March Collisions.

Figure 27: Heat Map of April Collisions.

7.5.2. Cluster Analysis Results

Ultimately, the models that identified five clusters in both March and April were selected. These models were chosen based on their usefulness and robustness of the clusters. The following describes the results for March and April as well as key findings, summarized in Tables 12 and 13. Lows are denoted in red, while highs are denoted in green. Figure 28 provides the March dendrogram.

7.5.2.1. March Analysis

In March, traffic patterns were typical, with the usual evening rush hour and congestion. Therefore, a higher number of hard braking and hard acceleration events occurred than in April. Segments averaged 349 hard braking events per mile and 129 hard acceleration events per mile—nearly two times and four times the averages of April, respectively. The cluster analysis identified the following five clusters:

- **Cluster 1:** Roadways with Low Crash Risk
- **Cluster 2:** Service Roads with Speeding
- **Cluster 3:** Service Roads with Elevated Crash Risk
- **Cluster 4:** Freeways with Elevated Crash Risk
- **Cluster 5:** Service Roads with Stop and Go Driving

The dominant characteristics in the March cluster analysis were hard braking and hard acceleration events. Clusters 2 and 5 exhibited higher than average hard braking and hard acceleration events, with Cluster 2 distinctly including a significant number of speeding events. Of the other clusters, Cluster 3 represents service roads with a higher number of historic crashes per mile; Cluster 1 includes both service roads and freeways with few historic crashes per mile; and Cluster 4 includes both service roads and freeways with a high number of historic crashes per mile.

Clusters 2, 3, and 4 represent safety concerns for TxDOT. Cluster 2 is noteworthy for its high speeding activity, indicating areas where TxDOT could implement speed management measures as well as coordinate with law enforcement. Cluster 3 includes roadways with higher percentages of fatal crashes and a higher percentage of incidents involving distracted driving. Cluster 4 has the highest number of historic crashes per mile and crashes involving a faulty evasive action. Most likely, these are instances where a rear-end collision occurs due to an abrupt change in the traffic pattern. Connected and automated vehicle technologies such as speed harmonization, hard brake warning, and automatic emergency braking may assist with reducing incidents in these high-crash corridors.

Table 12: March Results Summary

	MARCH				
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Hard Braking	Low	High	Average	Average	High
Hard Acceleration	Low	Average	Average	Average	High
Speeding	Low	High	Average	Average	None
Roadway Type	Mixed	Service	Service	Freeway	Service
Crashes per Mile	Low	Average	High	High	Low
Crash Severity	Moderate	Moderate	High	Moderate	Low
Contributing Factors	A	C, D, E	A	F	B

Table 13: March Results

Segmentation variable / Cluster	Overall	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Hard Braking (per mile)	349	**156**	623	294	472	**951**
Hard Acceleration (per mile)	129	**57.5**	281	151	101	**547**
Speeding (per mile)	82.1	7.39	**632**	48.8	45.3	**0**
Speed Limit	54.5	54.4	**50**	**50**	**59.1**	**50**

No. of Intersections	0.316	0.235	**1.27**	0.438	**0.0606**	0.333
No. of Lanes	2.88	2.75	**3.64**	**2.56**	2.91	3.33
Historic Crashes (per mile)	681	**199**	679	1200	**1260**	229
CS 1	0.12	0.03	0.21	**0.56**	0.03	**0.00**
CS 2	0.15	0.18	0.14	**0.05**	**0.18**	0.11
CS 3	0.15	0.15	**0.21**	**0.02**	0.20	0.03
CS 4	0.03	0.01	**0.00**	**0.16**	0.01	**0.00**
CS 5	0.43	0.36	0.44	**0.15**	0.58	**0.87**
CF A	0.09	0.06	**0.05**	**0.16**	0.11	0.15
CF B	0.02	**0.00**	0.02	0.02	0.03	**0.19**
CF C	0.01	**0.00**	**0.04**	0.01	0.01	0.01
CF D	0.02	0.01	**0.07**	**0.01**	0.02	0.02
CF E	0.00	0.00	**0.02**	0.00	0.01	**0.00**
CF F	0.00	0.00	0.00	0.00	**0.01**	**0.00**
Number of Observations	117	51	11	16	33	6
Proportion	1	0.436	0.094	0.137	0.282	0.051

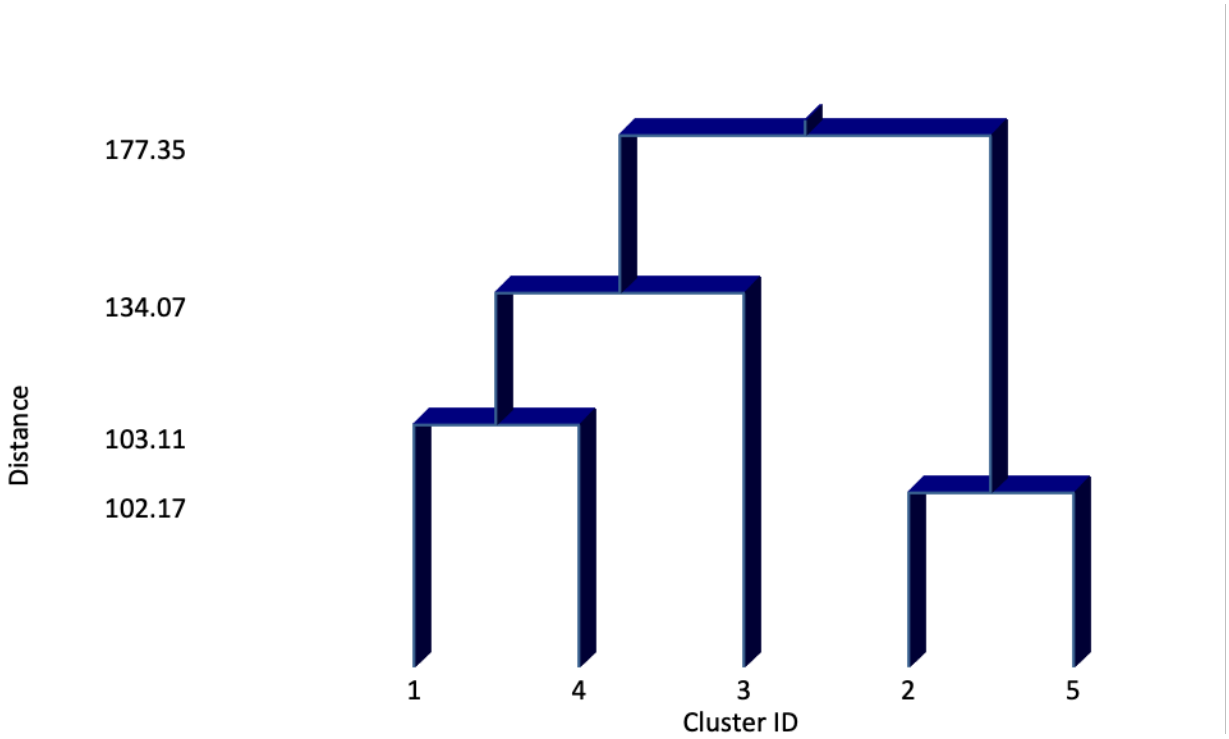


Figure.28: March Dendrogram.

7.5.2.2. April Analysis

In April, stay-at-home orders remained in place and people were traveling far less. As a result, fewer vehicles were on the roadway and people took advantage of the opportunity to travel at

higher speeds. This shift is reflected in the number of speeding events: an average of 219 speeding events per mile in April, as compared to the 82 speeding events per mile that occurred in March. When the cluster analysis was performed, the following five clusters were identified:

- **Cluster 1:** Typical Service Roads
- **Cluster 2:** Service Roads with Low Hard Braking and Low Crash Risk
- **Cluster 3:** Service Roads with Aggressive Driving
- **Cluster 4:** Service Roads with Low Hard Braking and Elevated Crash Risk
- **Cluster 5:** Typical Freeways

The dominant characteristics in the April cluster analysis were hard braking events and fatal crashes (Crash Severity 1) dominated. Clusters 2 and 4 exhibited lower than average hard braking and a higher percentage of fatal crashes, with Cluster 4 in particularly having a significantly high number of historic crashes per mile. Of the other clusters, Cluster 3 represents aggressive driving patterns with increased levels of hard braking, hard acceleration, and high speeding events; Clusters 1 and 5 exhibit more typical driving behaviors—breaking into subsets of service roads and freeways, respectively.

Clusters 3 and 4 represent safety concerns for TxDOT. In free-flow conditions such as those experienced in April, Cluster 3 identifies three service road segments where excessive braking, acceleration, and speeding occur: two consecutive service roads (NB) at the intersection of US 290, and one service road (SB) from 9th Street to 8th Street. Cluster 4 reveals five segments where historic crashes per mile are significantly elevated and high fatalities occur: service road (SB) from 4th Street to 3rd Street, service road (NB) from 7th Street to 8th Street, service road (NB) at the intersection of MLK Blvd., and two consecutive service roads (NB) from 5th Street to 7th Street. These segments are all located in the downtown area where pedestrian, cyclist, and vehicle activity is higher. Also, the primary contributing factor is "Following too closely," which could potentially be alleviated by CV technology and heads-up displays. Tables 14 and 15 summarize the April results; Figure 29 provides the April dendrogram.

Table 14: April Results Summary

	APRIL				
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Hard Braking	Average	Low	High	Average	Average
Hard Acceleration	Average	Average	High	High	Low
Speeding	Average	High	High	Low	Average
Roadway Type	Service	Service	Service	Service	Freeway
Crashes per Mile	Average	Low	High	High	Average

Crash Severity	Low	Moderate	Moderate	High	Moderate
Contributing Factors	B	A	D, E, F	C	A

Table 15: April Results

Segmentation variable / Cluster	Overall	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Hard Braking (per mile)	187	186	**74.6**	**741**	162	233
Hard Acceleration (per mile)	66.3	65.2	47.5	**591**	121	**15.2**
Speeding (per mile)	219	99.7	574	**1110**	**6.8**	99.4
Speed Limit	54.5	51.8	51.9	**50**	**50**	**64.4**
No. of Intersections	0.316	0.39	0.208	1	**1.2**	**0**
No. of Lanes	2.88	2.73	**2.58**	**4**	3	3.35
Historic Crashes (per mile)	681	672	**167**	911	**3010**	699
CS 1	0.12	0.07	0.18	0.07	**0.95**	**0.02**
CS 2	0.15	0.15	0.15	**0.20**	**0.00**	0.19
CS 3	0.15	0.15	**0.01**	0.27	0.05	**0.29**
CS 4	0.03	0.02	**0.08**	**0.00**	**0.00**	0.00
CS 5	0.43	**0.59**	0.03	0.47	**0.00**	0.50
CF A	0.09	0.09	0.08	0.08	**0.08**	**0.10**
CF B	0.02	**0.04**	**0.00**	0.01	0.02	0.01
CF C	0.01	0.01	**0.00**	0.02	**0.06**	0.01
CF D	0.02	0.03	**0.00**	**0.05**	0.01	0.01
CF E	0.00	0.00	**0.00**	**0.04**	0.01	0.00
CF F	0.00	0.00	**0.00**	**0.01**	0.01	0.00
Number of Observations	117	59	24	3	5	26
Proportion	1	0.504	0.205	0.026	0.043	0.222

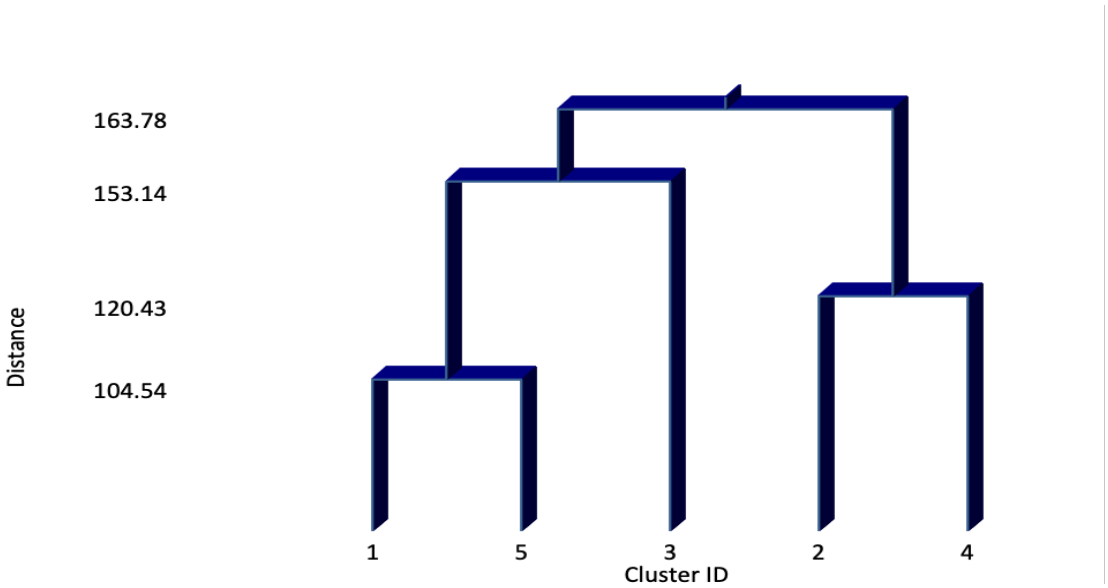


Figure 29: April Dendrogram.

7.5.2.3. Cluster Mapping

Once the cluster analysis was complete, the clusters were mapped to look for any patterns and understand what may contribute to the challenging driving segments. In March, Clusters 2, 3, and 4 were identified to have elevated risk (Figure 30). Cluster 2 (CL_2) is a collection of service roads with more aggressive driving behaviors, such as speeding, and a moderate crash risk. Clusters 3 and 4 (CL_3 and CL_4) both have elevated crash risk with Cluster 3 containing service roads that have relatively normal hard braking and hard acceleration counts and Cluster 4 containing freeway segments that have an increased number of hard braking occurrences, which could suggest heavier traffic. Cluster 3 would be of particular interest to investigate and understand why crashes are occurring here with relatively low speeding, hard braking, and hard acceleration, because it may suggest that a factor not considered in the cluster analysis is playing a role. Clusters 1 and 5 (CL_1 and CL_5) contain relatively low risk road segments that do not seem to have safety concerns to be addressed.

Figure 30: March Clusters (L: 1st to 15th St, R: Airport to US290).

Mapping the April clusters, results showed that some segments have shifted to a new cluster, which suggests that the changes in driving behavior that occurred during lockdown have some impact on road safety. For April, Cluster 3 and 4 are the high-risk road segments that have a higher volume of crashes (Figure 31). Cluster 3 (CL3) contains road segments that have a large number of hard acceleration and speeding events. These segments may be a good place for stronger enforcement effort and may benefit from coordination with law enforcement. Cluster

4 (CL4) has particularly high historic crashes with a large spike in crashes occurring where I-35 meets 11th Street. It would be interesting to investigate why these crashes are occurring, but this was out of scope for the case study. Clusters 1, 2, and 5 (CL1, CL2, and CL5) are all relatively low-risk road segments with Clusters 1 and 5 being average service road and freeway segments, respectively. Cluster 2 contains low-risk road segments, with fewer hard braking incidents and fewer historic crashes. It may also be helpful to evaluate these low-risk segments and see if there is anything to learn about why they tend to be safer.

Figure 31: April Clusters (L: 1st to 15th St, R: Airport to US 290).

Figures 32 and 33 show zoomed-in views of two sections of I-35 to illustrate some more detailed results. The section running from 1st Street to 11th Street reflects how some segments switched into new clusters and their risk assessment changed. Many of the stop-and-start segments transitioned to lower-risk segments with the reduced congestion of April. However, some of the higher-risk segments persisted and this may be helpful in identifying higher-risk segments that require more immediate attention. The US 290 Interchange is interesting because it contains a variety of clusters in a small area. The clusters suggest that the interchange does have some moderate- to high-risk segments that would benefit from attention. This aligns with the fact that many drivers find interchanges confusing and

challenging to navigate. Additionally, the main lanes of I-35 go from stop-and-go conditions in March to low-risk free flow in April, reflecting the reduced traffic on the roadways.

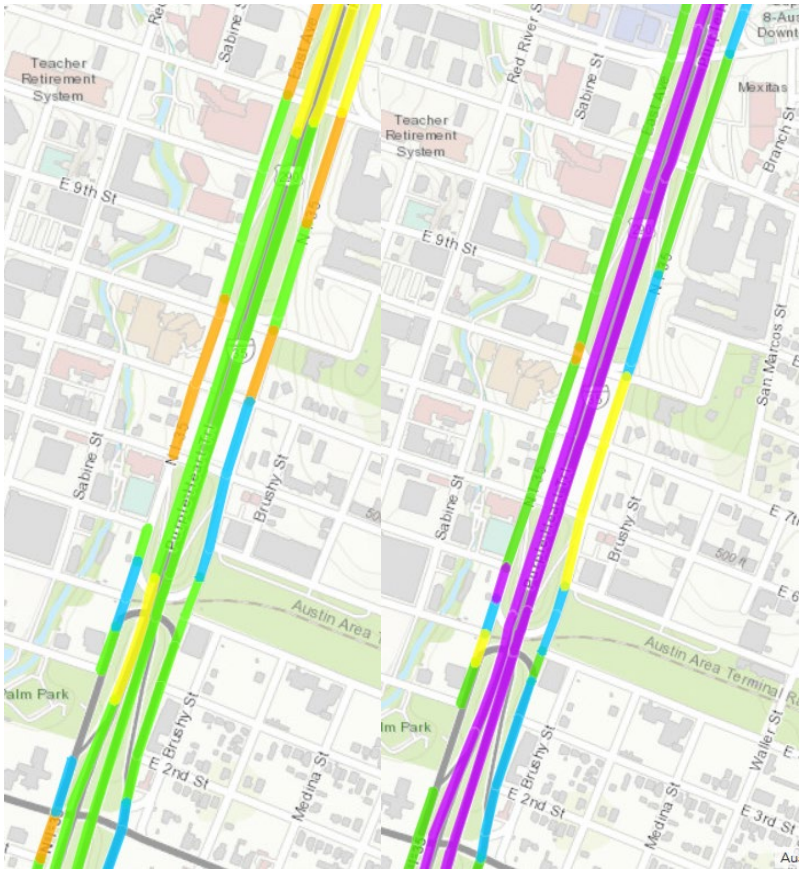


Figure 32: Clusters from 1st St to 11th St (L: March, R: April).

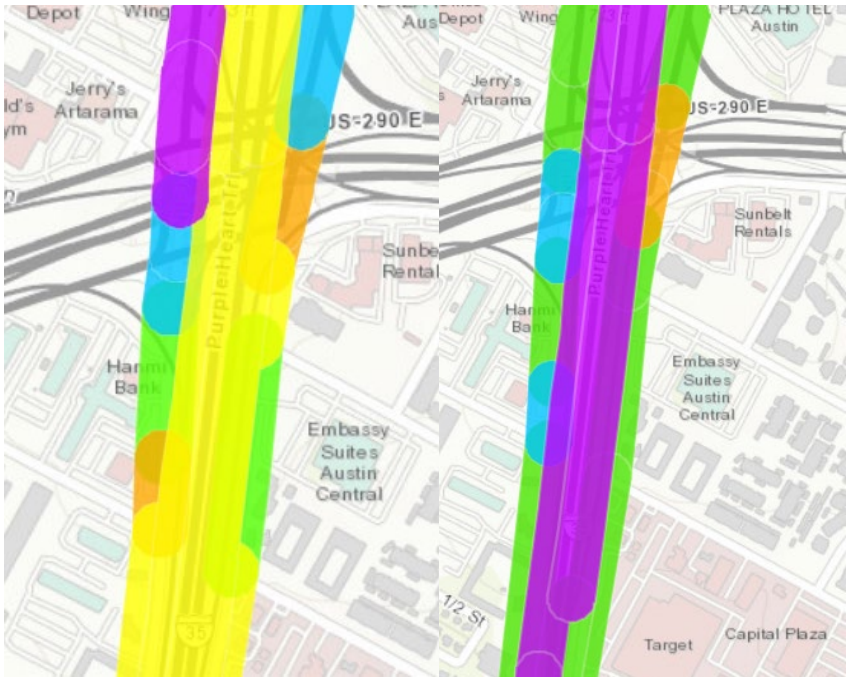


Figure 33: Clusters at US 290 Interchange (L: March, R: April).

7.6. Conclusions

In developing the prototype, several key findings are identified:

- Cluster analysis is a useful AI technique for classifying roadway segments based on their characteristics as well as identifying challenging driving environments. Clustering enables TxDOT to identify patterns of driving behavior (e.g., areas with high hard braking, high acceleration, and/or speeding events), compare service roads with freeways, and recognize major crash-contributing factors.

- CV data is a valuable source of information for providing insights into safety, planning, and operations. Event data that has already undergone processing to identify instances of hard braking, hard acceleration, and speeding is extremely useful to a transportation agency. For example, hard braking can identify bottlenecks, stop lights, or queuing from a crash. Hard acceleration can indicate signalized intersections that are close to one another or on-ramps to freeways. This type of information that is packaged in a way to be readily consumable can save a transportation agency significant time and money on computational resources. The data was found to be most useful in conjunction with other data sources, such as roadway geometry and crash reports.

- Results confirmed observations of lower traffic volumes and higher instances of speeding after stay-at-home orders were issued. Prior to the stay-at-home order, speeding typically occurred outside of downtown on some segments in the northern

stretch of the study area. Following the stay-at-home order, speeding events increased throughout the study area and most drastically near downtown.

- Crashes that occurred during the time period did not prove useful due to limitations in scope. Using crashes during the time period was not helpful due to low sample size, whereas the analysis of historic crashes was. Sometimes the crashes mapped to a centerline, and made it difficult to know the lane positioning and the NB or SB.

7.7. Opportunities & Recommendations

The research team identified several opportunities and recommendations:

Expand scope of the analysis: To make the model more robust, the research team recommends expanding the analysis to include a wider geographic area as well as time period. More information could be learned by including local arterials as well as other large facilities, such as MoPac or SH 71, for example. Additional research could also be performed to compare peak and off-peak times, weekday versus weekend, as well as track how traffic patterns evolved over the course of the COVID-19 pandemic.

Incorporate other data sources: There is an opportunity to enhance the cluster analysis with additional datasets, such as work zone information, weather, and other probe vehicle datasets.

Integrate CV data into safety, planning, and operations: This information provides significant insights and can be used to optimize TxDOT resources. As part of evaluating different CV data sources, TxDOT should determine what data frequency is necessary for its desired applications. For example, the Wejo data is generated once every three seconds, while standard CV data is generated at a rate of ten times per second. Comparing different frequencies and assessing what information is lost would be worthwhile to guide TxDOT's data investments.

Consider real-time data analysis: While this prototype was used to perform cluster analysis using historic data, a model could be developed for real-time data analysis that could detect driving behavior patterns and safety concerns as they happened. This information could enable TxDOT traffic management centers to communicate travel conditions to the public, coordinate with law enforcement and emergency responders, and integrate corridor management.

Refine speeding events: Speeding, as defined by Wejo, is an event triggered when a vehicle exceeds the 80-mph threshold. Some facilities in Texas, namely SH 130, have a speed limit of 80 mph or higher, meaning that the event threshold needs to be calibrated for those facilities.

Improve crash reporting: The crash reports in the CRIS database were not always complete or as accurate as a CV data source. In particular, the contributing factors were not always provided

by the reporting officer. This information can be incredibly valuable in identifying underlying causes of crashes and can assist agencies in achieving their goal to end all traffic fatalities. Furthermore, crash locations are often identified as where the incident is reported as opposed to where the incident occurred. Additionally, some crash locations were positioned on the I-35 centerline and it was unclear if they occurred on a northbound or a southbound segment. CV data presents an opportunity to map with high accuracy when and where crashes occur, as well as provide additional context on contributing factors.

Chapter 8. Travel Patterns Using Supervised Learning

The study described in this chapter explores the use of machine learning methods to understand changes in trip-making behavior that occurred during the COVID-19 lockdown. A dataset of trip events registered by CVs was used for this study. The trip data was aggregated to determine the number of trips originating and terminating in each census-block-group on an average weekday before the lockdown and on an average weekday after the lockdown. A variety of supervised machine learning techniques were employed to model the relationship between the trip-making behavior and socio-economic characteristics of census-block-groups. The developed models were then used to make inferences on the changes in trip-making patterns during the lockdown.

8.1. Data

8.1.1. Wejo CV Data

The primary dataset used for the study was the Wejo CV dataset. This dataset contained records of all the trip events registered by a sample of CVs in the Austin metropolitan area over a one-week period before the lockdown and a one-week period after the lockdown. The data collected on the weekdays between March 2 and March 6 of 2020 was used to analyze trip-making before the lockdown. Data collected between April 6 and April 10, 2020, was used to analyze trip-making during the lockdown. The trip events that were registered include change in seatbelt status, speed crossing a certain threshold, acceleration crossing a certain threshold, and change in journey status. Each event record was associated with location coordinates, time, device ID, and journey ID. Only the events pertaining to change in journey status were of interest for this study. The JOURNEY: START event was registered whenever the vehicle's ignition was turned on and the JOURNEY: END event was registered when the ignition was turned off. These events could be used to identify the beginning and end of trips. Some cases of duplicated records were identified in the dataset and removed before further processing.

8.1.1.1. Generating the Dataset of Trips

The JOURNEY: START and JOURNEY: END events that had the same journey ID were used to identify the beginning and end of individual trips. The trip origin, destination, start-time, and end-time could be extracted from the time and location data associated with these journey events. Multiple start and end events had been recorded for a few journey IDs, possibly because of errors in the data collection process. An inspection of these journey IDs revealed that in most cases only one pair of the journey start and end events could be real, as the other

pairs would result in unreasonably high or unreasonably low travel times. Therefore, for these journey IDs, only the pair of journey events that would result in a travel time that is closest to 30 minutes was used.

The number of trips in the dataset after executing the aforementioned data processing steps was 1,278,738. The spatial and temporal distribution of the trip origins were checked. It was noted that the number of trips tended to spike around auto-repair shops and vehicle showrooms. This spike was probably caused by the vehicle repair and testing activities that occur at these locations and not because of actual trip-making. To eliminate these invalid trips, the dataset was further filtered to include only the trips where the origin and destination points are at least 400 meters (~¼ mile) apart in terms of straight-line distance. This step would also remove other trips made for trivial reasons, such as changing parking locations. Few genuine trips would also be removed where the trip origin and destination happened to be nearby. However, since actual short-distance trips by car are relatively uncommon, it should not significantly impact the analysis.

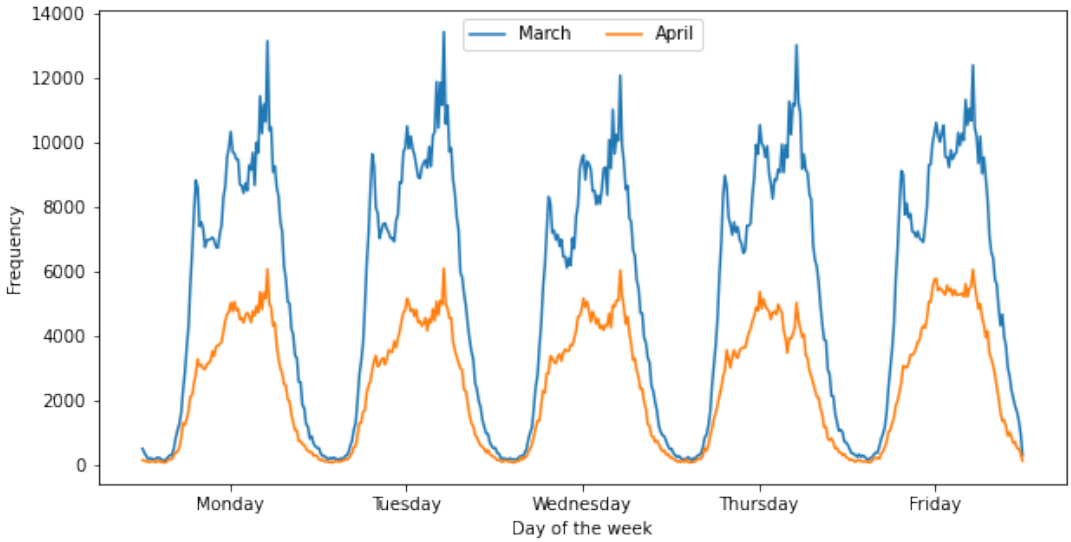


Figure 34 Trip Frequency (trips/h) Plotted at 15-minute Intervals for March and April

The filtered dataset had a total of 996,908 trips. Of these, 659,564 of the trips were made in March and the rest (307,344) were made in April. The temporal distributions of the trips in March (pre-lockdown) and the trips in April (during lockdown) are provided in Figure 34 In the plots for the month of March, the three peaks that occur in each day approximately correspond to the times of 7:00 AM, 12:00 noon, and 5:00 PM respectively. As expected, a much smaller number of trips were generated in April. Interestingly, the AM peak is much less prominent or not present in April. The spatial distribution of trip origins is shown Figure 35. Once again, it is clear that the number of trips made in April is much lower than the number of trips made in March. The concentration of trip origins seems to be highest in the Austin

downtown area and around The Domain.

8.1.1.2. Block-Group Level Aggregation

The dataset of census-block-group level socio-demographic characteristics was obtained from the 2018 American Community Survey (ACS) Database (US Census Bureau, 2021). The ACS data from 2018 was used because this was the most recent dataset available at the time the analysis was conducted. Only the data from the counties of Bastrop, Caldwell, Hays, Travis, and Williamson were selected because all the trips occurred within these counties. The counties consisted of 967 census-block-groups, henceforth referred to simply as block-groups. The shapefile of the block-groups was also obtained, and the trip origin and destination locations were mapped on to the block-groups. The mapped data was then aggregated to determine the total number of trips generated from each block-group during March and April. The number of trips in the block-groups were divided by the duration of five days to compute the frequency of trips generated in the block-groups. These frequencies are plotted in Figure 36. The absolute and percentage change in trip frequencies is displayed in Figure 37. The block-groups that produced fewer than 50 trips during the one-week period in March has been grayed out as the percentage change in trips in these block-groups would be unreliable.

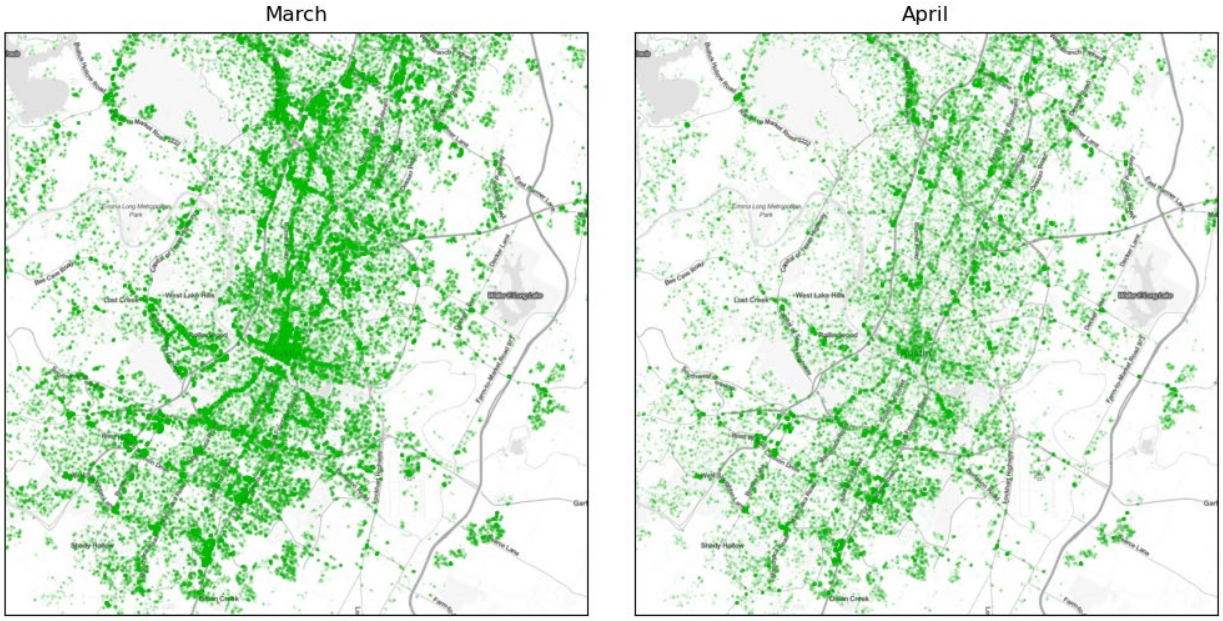


Figure 35. Locations of Trip Generation.

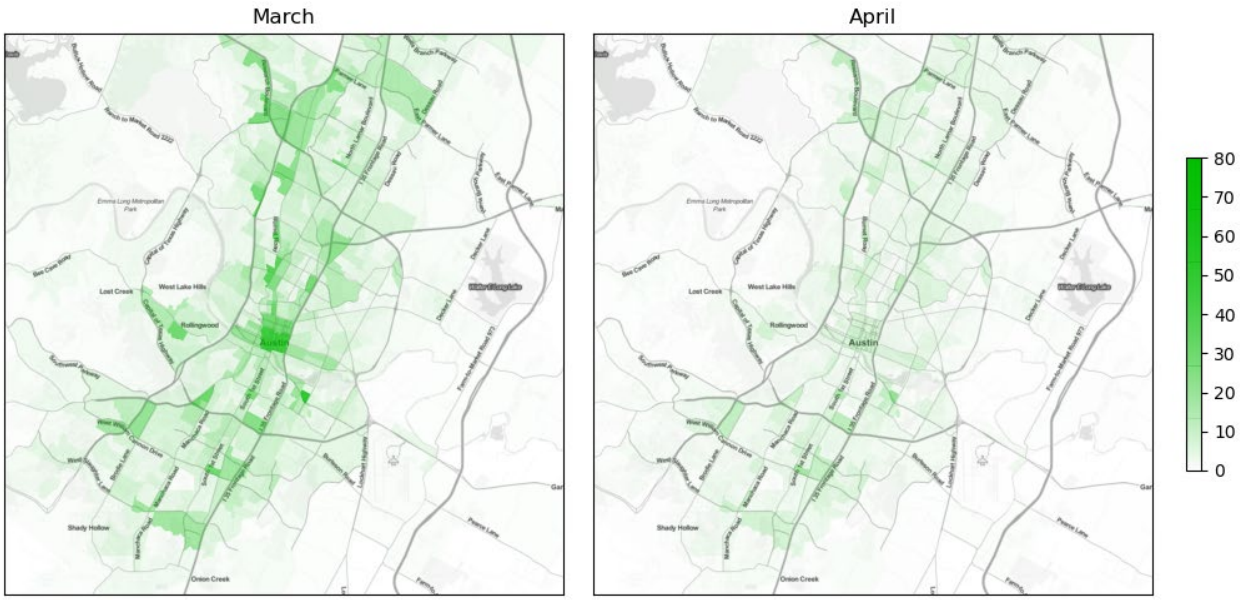


Figure 36. Census-Block-Group Level Trip Frequency (h-1).

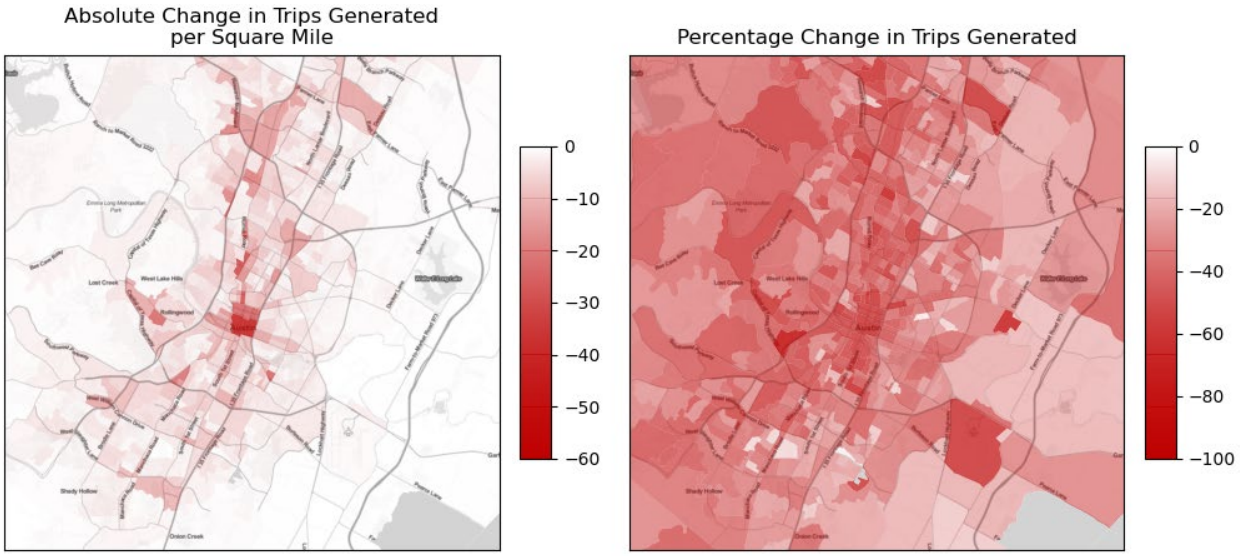


Figure 37. Change in Trip Frequency per Block-Group – Absolute (Trips/mi^2/h) and Percentage.

The socio-demographic data from the ACS database was supplemented with the data on employment locations in Austin. The employment dataset was obtained from the Capital Area Metropolitan Planning Organization. The dataset consisted of locations, employment type, and number of employees of businesses in Austin. This employment dataset was mapped onto the block-group shapefile and aggregated to obtain the total number of basic, retail, service, and education employment opportunities in each block-group.

8.2. Analysis Methodology

The strategy for analysis involved the development of models that can predict the number of trips generated in any block-group based on the socio-demographic and employment characteristic of the block-group. Separate models would be developed for March and April. The sensitivity of the models to socio-demographic and employment characteristics would help us understand the effect these characteristics have on trip generation. The target variable or dependent variable of the models is the frequency of trips generated in a block-group. The number of households having different household sizes and the number of employment opportunities of each category were used as predictors or independent variables.

Models of this type, where the travel demand of an area is modeled directly as a function of factors such as socio-demographic characteristics, employment opportunities, and built-environment characteristics, are referred to as direct demand models (Ortúzar and Willumsen, 2011). The modeling framework most commonly used for direct demand analysis is the multivariate linear regression model. The multivariate linear regression model, henceforth referred to simply as linear regression, provides the advantage of an intuitive model structure where the estimated parameters directly indicate the sensitivity of the target variable towards the predictors. A disadvantage of linear regression is that the model is inflexible in the sense that if the predictors and target variable do not have a linear relationship, the prediction accuracy tends to suffer. This inflexibility also causes linear regression models to perform poorly when outliers are present in the dataset. Outliers are records where the value of a predictor variable or the target variable is significantly different from the values in all the other records in the dataset.

Recently, supervised machine learning methods have been gaining more traction in direct demand modeling. These methods are generally more capable of incorporating nonlinearities in the relationship between the predictor variable and the dependent variable. As a result, machine learning methods are generally more accurate than the Linear Regression models (Ding et al., 2019; Yan et al., 2020). A disadvantage of the more complex non-linear machine learning methods is that the model structure is less intuitive, and as a result, making inferences on the effects of changes in the predictor variables is more challenging. In this study, we develop models using the Linear Regression approach as well as two machine learning approaches: random forest and multilayer perceptron. The accuracy of each of these models was evaluated in terms of their coefficient of determination (R^2). The sensitivity of the models to block-group characteristics was also investigated.

8.2.1. Measure of Fit

The coefficient of determination or R^2 is the proportion of variance in the target variable that is explained by the predictors. Consider N records of the outcome variable indexed as $y_1, y_2, y_3, \ldots, y_N$. Let the predicted outcome for these records be $\hat{y}_1, \hat{y}_2, \hat{y}_3, \ldots, \hat{y}_N$. Then R^2 is computed as follows,

$$R^2 = 1 - \frac{SSR}{SST} \tag{1}$$

$$SSR = \sum_{n=1}^{N} (y_n - \hat{y}_n)^2 \tag{2}$$

$$SST = \sum_{n=1}^{N} (y_n - \mu)^2 \tag{3}$$

where μ is the mean of the target variable, *SSR* is the sum of squared residuals, and *SST* is the sum of squared totals.

8.3. Dataset Preprocessing

The block-group level trip frequency data was further preprocessed. These preprocessing steps would enable the models to produce more accurate and more generalizable results.

8.3.1. Removing Outliers

The initial analysis using the block-group level trip frequency dataset resulted in poor prediction accuracies from the linear regression model. On further investigating the dataset, few outliers were identified. A block-group covering a portion of the downtown area had a much higher number of service employment opportunities than the other block-groups. Similarly, a block-group partially covering the University of Texas at Austin main campus area had more than 20 times the number of education employment opportunities in the next highest block-group. These block-groups and two other block-groups with a significantly high number of employment opportunities were removed from the dataset. In practice, these locations would be marked as special generators and would not be handled in models like the linear regression model in a conventional manner. The non-linear machine learning methods are usually more robust to outliers. Nevertheless, these block-groups were not included in the analysis for a fair comparison between the models.

8.3.2. Removing Biased Records

The trip data was collected only from a sample of all the vehicles in Austin. If the sampling is biased towards any of the predictors, the sensitivities computed with respect to that predictor

may not be generalizable to the case when all vehicles are considered. For example, if the proportion of retail employees whose vehicles are included in the sample is much higher than the proportion of other sectors' employees whose vehicles were sampled, the analysis would likely overestimate the effect of retail employment on trip-making. This is because the trips made from retail employment locations are more likely to be recorded in our dataset than the trips made from other locations.

In the dataset, one block-group had one of the major offices of the OEM of the vehicles for which the data was collected. For this block-group, the random sampling assumption would not hold because a disproportionately large share of the employees in these offices owned the vehicles for which the data was collected. This block-group was removed from the analysis so that the oversampling of trips from this block-group would not bias the model estimates.

It needs to be noted that, at this stage of the study, removing the block-group with the OEM's offices was the only step taken to avoid sampling bias. A more thorough investigation on the exact extent of bias in each of the block-groups may be warranted if these models are to be developed further.

8.3.3. Removing Block-Groups with Low Trip Counts

The trip frequencies in some of the block-groups were zero or quite low. The data on trip origins from these block-groups would not be reliable for assessing the change in trip-making behavior before and after the lockdown. This is because even a small change in the number of trips at these block-groups would represent a large change in terms of percentage. Therefore, the block-groups where the number of trips made on an average day in March was 20 or less was removed. After all the preprocessing steps, 792 block-groups remained in the dataset.

8.4. Training and Testing Method

An issue faced when developing non-linear machine learning models with a large number of degrees of freedom (high flexibility) is that the model may precisely fit the dataset used for training, but the trained model may not produce accurate results for records that were not used for training. This phenomenon is referred to as overfitting. Since machine learning models are prone to overfitting, it is not recommended to test the accuracy of the model with respect to the dataset used for training. Therefore, the complete dataset was split into two groups in an 80:20 ratio. The larger subset of data containing 80 percent of the records was set aside for training the models. The model accuracy was tested on the other subset containing 20 percent of the records.

To avoid overfitting, the degree of freedom used by the machine learning model needs to be high enough that the nonlinearities in the relationship between the target variable and predictors are captured but not so high that it fits to the noise in the training dataset. The degrees of freedom of a machine learning model are controlled by special types of model parameters called hyperparameters. These hyperparameters can be adjusted to provide the optimal amount of flexibility to the model. The process of setting the optimal hyperparameters is referred to as hyperparameter tuning. Hyperparameter tuning is essentially a trial-and-error approach to find the best set of values for the hyperparameters. In the hyperparameter tuning process, the training dataset is further split into two parts, usually in an 80:20 ratio or 2:1 ratio. The larger of these subsets is used for training the model with a specific set of hyperparameter values and the accuracy of the resulting model is evaluated with respect to the smaller subset. The hyperparameter values that produce the best results in this process are identified as the "tuned" hyperparameters values. This process of splitting and evaluation may be done several times with the same set of hyperparameter values but with different sets of records in each of the split groups. After hyperparameter tuning, a model with the tuned hyperparameters is estimated using the full training dataset; Figure 38 illustrates the data split.

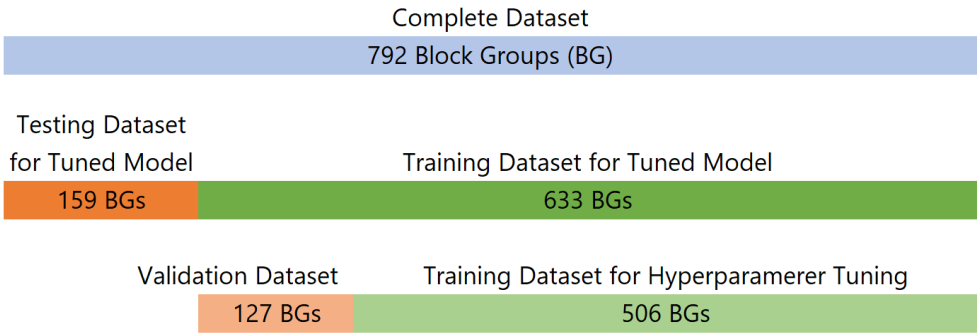


Figure 38 Dataset Split into Training, Testing, and Validation Datasets.

8.5. Modeling Frameworks

8.5.1. Linear Regression

In a linear regression model, the target variable is expressed as a linear function of the predictors. Consider a vector of the target variable for the N records: $y_1, y_2, y_3, \dots, y_N$. Let the vector of predictors corresponding to the nth record be $x_{n1}, x_{n2}, x_{n3}, \dots, x_{nI}$. Then, in linear regression, the value predicted for the nth dependent variable will be of the form,

$$\hat{y}_n = \beta_0 + \sum_{i=1}^{I} x_{ni}\beta_i \tag{4}$$

where $\bar{\beta} = (\beta_0, \beta_1, \beta_2, \dots, \beta_I)$ are the vector of parameters to be estimated during model training. These parameters are determined by minimizing an objective function that is

correlated with the error in predicting the target variable. The ordinary least squares (OLS) method is among the simplest methods used for estimating the parameters. In this method, the estimated parameters are such that the sum of squared errors from prediction is minimized. That is,

$$\bar{\beta} = \underset{\beta}{\operatorname{argmin}} \left(\sum_{n=1}^{N} \left(y_n - \left(\beta_0 + \sum_{i=1}^{I} x_{ni} \beta_i \right) \right)^2 \right) \tag{5}$$

8.5.2. Decision Tree Regression

In a decision tree model, the records in the training dataset are divided into groups based on a set of yes-or-no conditions on the predictor variables. The conditions will be such that the set of records that fall within the same group will have similar values for the target variable. The set of yes-or-no conditions that are used to group the records can be represented in a tree format and hence the name decision tree model. Each internal node of the tree represents a comparative condition on a predictor of the dataset and will have two child nodes corresponding to whether the condition evaluates to true or false. The terminal nodes or leaf nodes of the decision tree hold the value to be predicted for any record that reaches the node. This value will be the mean of all the records in the training dataset that reach the node. Since the use of a large number of conditions on the predictors can result in overfitting of the decision tree to the training dataset, the maximum depth of the tree (d) and the minimum number of records from the training dataset that should reach an internal node (s) are considered as the hyperparameters of the model. An example of a decision tree is visualized in Figure 39.

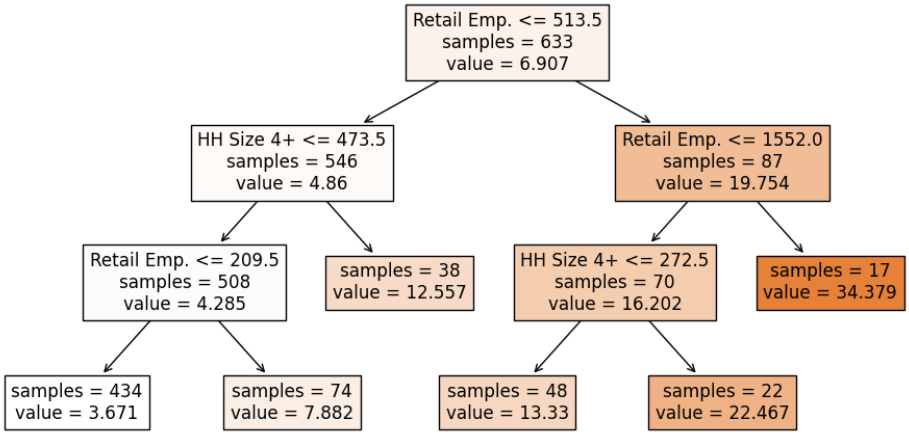


Figure 39 Example Decision Tree.

Note that model results from the decision tree model are not included in the results section. Instead, results from the random forest regression model (explained in the next section), which is a more generalized version of the decision tree model, are provided.

8.5.3. Random Forest Regression

The decision tree model is prone to overfitting even when the hyperparameters are appropriately tuned. An extension (ensemble version) of the decision tree model that generally produces better predictions is the random forest regression model. In random forest regression, multiple decision trees are trained on random samples drawn with replacement from the training dataset. The random samples have the same size as that of the original training dataset. The final prediction will be the average of the prediction made by each of the component decision trees. The accuracy of random forest regression tends to increase with the number of component trees, along with computational cost. In this model, each random forest model was generated with 200 component decision trees. The hyperparameters of the decision tree model—the maximum depth of the trees (d) and the minimum number of records reaching an internal node (s)—are also hyperparameters of the random forest model. An additional hyperparameter used in random forest is the maximum number of predictors allowed per tree (f). The number of predictors that can be used for branching by any one decision tree in the random forest is limited to f. The tuned values for the hyperparameters were $d = 10$, $s = 6$, $f = 3$ for March and $d = 8$, $s = 12$, $f = 5$ for April.

8.5.4. Multilayer Perceptron Regression

The multilayer perceptron regression model is a type of neural network model. This model is structured by stacking together multiple computational units called perceptrons. The perceptron is a computational unit that accepts an input vector $x_1, x_2, x_3, \dots, x_I$ and applies a nonlinear monotonically increasing function over it to produce the result. The computation that occurs in a perceptron can be expressed as,

$$a = f(w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_I) \tag{6}$$

where a is the output from the perceptron, and f is the nonlinear function. $w_0, w_1, w_2, \dots, w_n$ are parameters that need to be estimated. In multilayer perceptrons and neural networks, several of these perceptrons are stacked together in layers such that the output from the perceptron in one layer is passed as input to the perceptrons in the next layer. The final output of the model is the output from the perceptrons in the last layer of the network. The number of layers (hidden layers) between the input layer and the output layer and the number of perceptrons present in each layer are hyperparameters. Neural networks in general have a large number of hyperparameters and parameters, which allows the networks to map the

relation between an extremely diverse range of inputs and outputs. What has been mentioned above is only one possible way of configuring a neural network. In this network we use only one extra layer between the input and out layer. The number of perceptrons in this layer is kept the same as the number of predictors (inputs). The activation function f is set as the sigmoid function σ.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{7}$$

The early-stop approach was used to avoid overfitting. The training of neural networks involves progressively updating the parameters such that the resulting network more closely matches the relationship between the predictors and the target variable. In the early stop approach, after every few iterations, the accuracy of the neural network is evaluated with respect to a dataset that is not used for training called the validation dataset. Once the predictive accuracy on the validation dataset stops increasing, the training is stopped.

8.6. Model Results and Discussion

Separate prediction models were estimated for March and April. The coefficients estimated using the Linear Regression model are given in Table 16. These coefficients represent the sensitivity of the Linear Regression model to the change in corresponding predictors. The standard errors of the estimated coefficients are also provided. The coefficients of all the predictors except that of Education Employment in April are different from zero at a 95 percent confidence level. The coefficient estimated for a predictor can be interpreted as the extra trip frequency that would be "observed" for a unit increase of that predictor. Note that this is not the actual number of extra trips that would be caused by a unit increase of the predictor because the estimation is based on only an observed fraction of all the vehicles. If we assume that the fraction of observed vehicles is the same for each of the household categories and for each of the employment categories (that is, if the random sampling assumption holds), the estimated coefficient will be directly proportional to the actual number of trips made by that category.

Table 16: Coefficients Estimated Using Linear Regression and the Percentage Change in Coefficients between March and April

Block-Group Feature / 100	March		April		% Change
	Coeff.	Std. Err.	Coeff.	Std. Err.	
(Intercept)	-0.2057	0.237	-0.0388	0.138	--
Household Size 1 or 2	0.1371	0.046	0.0733	0.027	-46.54%
Household Size 3	0.6536	0.195	0.4306	0.114	-34.12%
Household Size 4+	1.0274	0.094	0.5712	0.055	-44.40%
Basic Employment	0.1269	0.037	0.0685	0.022	-46.02%

Block-Group Feature / 100	March		April		%
	Coeff.	Std. Err.	Coeff.	Std. Err.	Change
Educ. Employment	0.5078	0.122	0.0895	0.071	-82.37%
Retail Employment	1.0296	0.032	0.4137	0.019	-59.82%
Service Employment	0.1363	0.017	0.0247	0.010	-81.88%

The coefficients for the household size variables show the expected pattern of being higher for larger households in both March and April. Among the employment variables, retail employment produces the largest number of trips. The percentage change in coefficients provides an estimate of the percentage change in trip generation in each of the categories after the lockdown. The highest percentage reduction in trip rate is observed in locations with more service and education opportunities. Among the household categories, the percentage reduction in number of trips was higher for household with sizes 1 or 2 and households with size greater than 4. Despite the uneven percentage reduction in trips, larger households continued to make more trips than smaller households.

As described earlier, the machine learning models do not have parameters that are as easily interpretable as the parameters of the linear regression model. However, inferences can be drawn from the predictions made by the model. To understand the sensitivity of the machine learning models, an average block-group was considered where the value of each of the predictors was set as the average value over all the block-groups. Then, to understand the sensitivity of a predictor x_i, the value of x_i was iterated over a range of values between 0 and the 99th percentile value of the predictor and the trip frequency was predicted in each iteration. Plotting this predicted frequency with respect to the values of the altered predictor would give a sense of the effect the predictor has on trip generation in an average block-group. The sensitivity plots generated in this manner are provided in Figure 40.

As expected, generally the trend seems to be that as the number of households and the number of employment opportunities increase, the number of trips generated also increases. Larger households generated more trips both before and during the lockdown. Similarly, retail locations generated the most trips before and during the lockdown. During the lockdown, among the employment categories, only retail employment locations seem to contribute to a reasonable extent to the number of trips made (displaying a higher slope than other employment categories). Increase in other employment opportunities barely resulted in any higher number of trips. This could be because individuals employed in the education and service industries were most allowed to work from home and the extent of basic employment activity was considerably reduced.

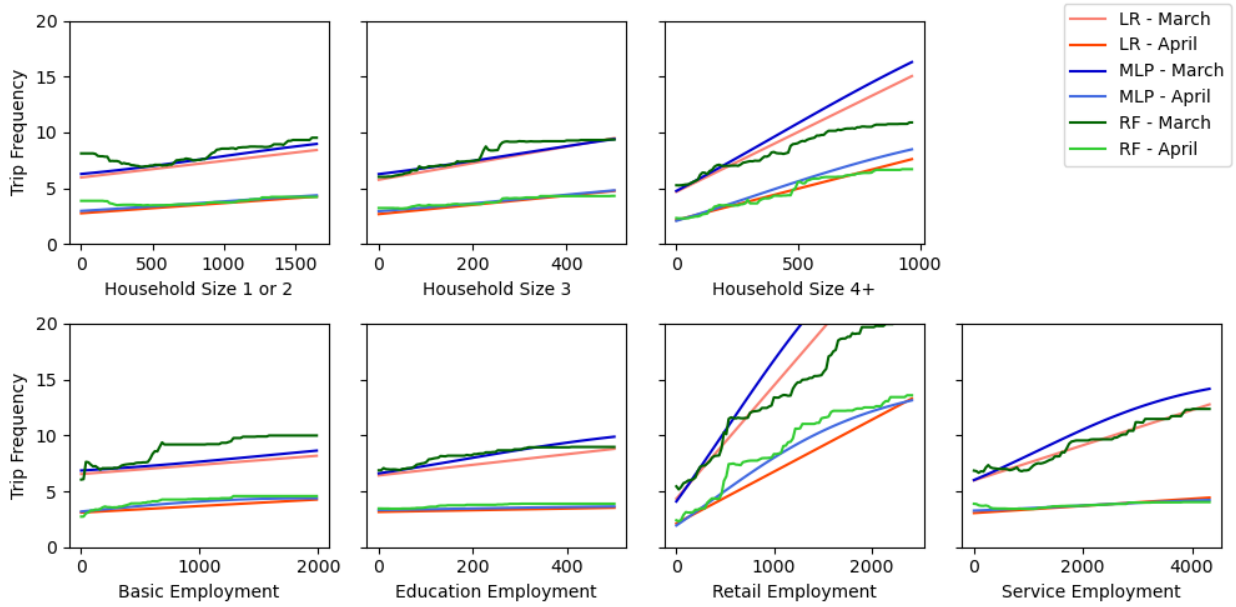


Figure 40 Effect of Predictor Variables on an "Average" Block-Group.

Among all the models, the random forest model seems to be the most flexible based on the rapid changes in slopes exhibited by its predictions. This also makes the random forest model likely to overfit to the training dataset. The predictions made by the multilayer perceptron and the linear regression model seem mostly similar except for the retail employment and service employment cases. The predictions made by the random forest method seem to diverge from the predictions made by other models, especially at large values of retail employment and household size. This could be because the random forest model and the other models use different approaches to make predictions for records outside of the range of records in the training dataset. While the Random Forest model, by design, cannot produce predictions that are higher than the maximum for similar records present in the training dataset, the linear regression model and multilayer perceptron model can extend the predictor-target relationship outside the range of the training dataset. An oddity observed is that the random forest predicts relatively higher frequencies for low number of households of sizes 1 and 2. While understanding the exact reason for this deviation would require further investigation, such issues are commonly caused because of the presence of other effects or factors that are simultaneously related to the number of small households and the trip frequency.

To understand the accuracy of the models and the extent of overfitting, the R^2 of the models were computed with respect to the training dataset and testing dataset. The computed R^2 values are displayed in Figure 35. The model that produced the best fit was the random forest model based on the R^2 values with respect to the testing dataset. The multilayer perceptron model performed similarly well while the fit of the Linear Regression model was slightly worse especially for the month of April. Note that the equivalent performance of the Linear

Regression model, when compared to the other models, was possibly partly because of the removal of outliers in the preprocessing step.

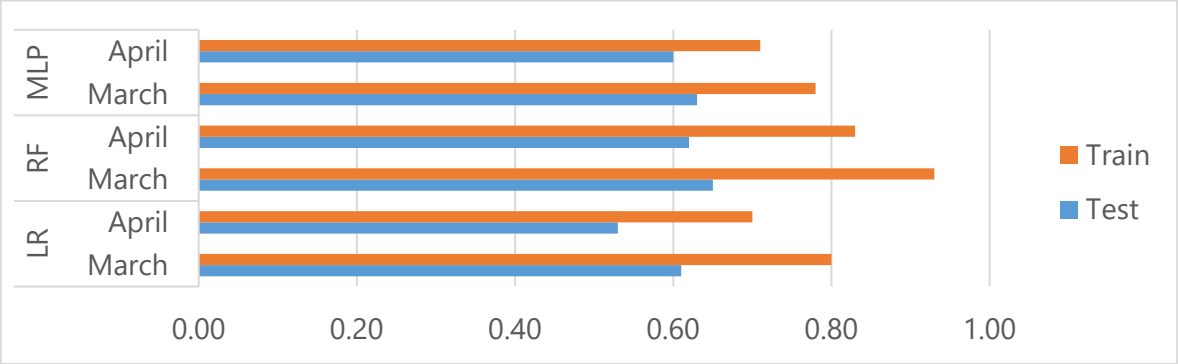


Figure 41 R^2 for Estimated Models with Respect to Training and Testing Datasets.

8.7. Conclusions

The use of machine learning techniques for the implementation of a direct demand model was discussed. The methods for extracting and aggregating the data from CV event logs were explained in detail.

The highest percentage reduction in the number of trips was observed at location with higher service and education opportunities. During lockdown, among the employment categories, only retail employment locations seemed to make a reasonable contribution to the number of trips. Households also made a reduced number of trips during the lockdown. Households that are larger tended to produce more trips than smaller households both before and during the lockdown.

Overall, more complex machine learning methods performed better than linear regression when there were outliers in the dataset. Once these outliers were removed, the performance of the linear regression model was similar to that of other machine learning models. This suggests that if the goal is to develop a “one size fits all” model that is robust to outliers, the random forest approach is recommended. However, a drawback of the random forest model that must be kept in mind is that the model may not produce reasonable predictions for records outside of the range of records that the model was trained on. The multilayer perceptron model is not recommended even though it performed almost as well as the random forest model because training a multilayer perceptron is usually more involved and therefore more challenging to automate. The analysis of changes in trip-making patterns after the lockdown is only an example use case of direct demand models developed using CV data. More generally, the development and maintenance of such models based on a live stream of CV data will allow for the rapid analysis and identification of changes in travel patterns that occur in any novel situation.

Chapter 9. Traffic Signal Timing Using Reinforcement Learning

Chapter 9 investigates the application of reinforcement learning (RL) techniques to the traffic management scenario. In particular, the goal of integrated corridor management (ICM) is in focus, in hopes of enabling advanced control techniques that can respond to observed roadway conditions faster and more effectively than current practices, which focus primarily on human design choices. Given the myriad data sources available to transportation management centers (TMCs), we assume that a plausibly accurate representation of a roadway network's congestion conditions can be developed. Furthermore, standardized mechanisms of traffic control exist, including the National Transportation Communications for Intelligent Transportation Systems Protocol (NTCIP), thus providing a unified framework for influencing roadway traffic levels.

We aim to leverage the available technology to develop an RL agent that can observe roadway conditions through the data sources identified in other sections of this project and respond appropriately once trained in a simulated environment. This arrangement allows for safe testing of potentially dangerous control policies in a sandboxed environment as the agent learns to function properly. We next discuss the common components of a RL agent model, including how each component ties to the experiment at hand. In this chapter, we discuss the initial proof-of-concept model that was developed, including the scenario generation technique and development of a training environment. In addition to the results from this model, we further detail in Section 4.3 a series of alterations made to the simulated environment to decrease the artificiality associated with the training environment, as well as the implications of those changes. We conclude with a discussion of further steps we aim to undertake in improving the RL agent.

9.1. Components of Reinforcement Learning

RL agents are typically represented as consisting of five key components—three under the influence of the agent (the environment, the performance element, and the learning mechanism), and two independent components (the problem generator and the evaluation standard).

9.1.1. Environment

TMCs have at their disposal a series of sensors through which they observe the state of traffic on roadways, as well as multiple actuators that allow them to exert influence on the roadways' traffic state. Current practice utilizes inductive loop and other magnetometers, traffic cameras, probe travel time data, and CV data to provide TMCs with an understanding of the state of traffic; operators can then adjust actuators such as traffic signal timings, variable lane usage

signs, and toll rates to drive traffic to more optimal or safer routes. These sensors and actuators thus define an environment in which an RL agent can operate, moving throughout the state space via the actuators available to it. However, without further structure, the agent will be unable to make intelligent decisions as to how to move about the state space.

9.1.2. Performance Element

In RL environments, the performance element chooses what next action(s) should take place based on the knowledge it has gleaned through its training. This policy determines the perceived optimal action(s), and by repeated interaction with the environment, the agent's performance will approach true optimality. Since this element is independent from the learning element, the RL agent can evolve to understand complex relationships between its choices in actions and the results that arise from them. This, therefore, distinguishes learned behavior from simpler behavior more akin to reflexes in humans.

9.1.3. Learning Mechanism

Given the extreme combinatorics of traffic states across an entire network, it becomes necessary in the traffic management scenario for an agent to learn more through practice than through brute-force sampling of every policy option. This is because time will be a limiting factor in searching the simulated state spaces. For such a complex problem, it is common to see RL strategies such as Q-learning and SARSA employed. These methods allow an agent to gain direct feedback on the evolution of the environment following the agent's action choice. Q-learning, an off-policy technique, recommends a policy but is indifferent as to whether the policy is followed. In contrast, SARSA learns based not only on the prior state and the recommended action(s), but also from the action(s) that are, in fact, implemented, and the state that results therefrom. This ability to reflect on the actual behavior is termed an on-policy learning method. A multitude of other learning algorithms exist that build on these fundamentals, and we review a limited exploration of these mechanisms in our experiments.

9.1.4. Problem Generation

In nearly all nontrivial environments, RL performance depends on a balance between the goals of the agent—the drive to perform optimally in the current state given the available knowledge, as well as the desire to find new strategies that may offer better rewards. This is referred to as the exploration-vs.-exploitation balance. An independent problem generator can randomly select instances where new action settings should be used. By choosing a fixed frequency of these random deviations from the prototypical policy, an ε-optimal selection strategy is defined, allowing for a variable rate of exploring new options relative to the policy's predetermined actions and other policy choices available. Another approach, selecting

randomly based on the amount of prior explanation from a given state, allows for an awareness of the "known unknowns" in an RL agent's policy design. Other design options include basing the exploration rate on the number of times a given state scenario (or an approximately identical one) has been encountered.

9.1.5. Evaluation Standard

The independent critic is possibly the most important module for an RL agent to truly develop optimal behavior, as it provides feedback to the learning element describing the agent's performance based on the sensor data available to it. For the agent to understand the benefits and costs of its actions, it must be able to receive external metrics on the value of its performance. The agent will then seek out the policy decision(s) that maximize the reward from such a critic, regardless of unmeasured side-effects thereof. It is necessary for the feedback element to be defined outside the learning mechanism, as agents must not be able to arbitrarily determine the value associated with states in a sycophantic attempt to further the reward it is given. In the transportation environment, multiple metrics exist for evaluating performance, including total system travel time, levels of service (LOSs), and many others. We later discuss the implications of the performance metrics chosen for our experiments and illustrate just how critical an apt reward evaluation mechanism is in designing RL agents for ICM.

9.2. Initial RL Model

To prove the applicability of the RL mindset to the traffic control problem, a toy model was developed that gave an RL agent control over an intersection's operations. This section details the resources available to the RL agent, the training environment to which it was connected, and the results of the initial experiments therefrom. In the next section, we detail improvements made over the initial design and the implications these modifications had for agent performance.

9.2.1. Scenario Development

The initial test scenario models a pair of frontage roads along a freeway, as well as a cross street that intersects the frontage roads at a pair of signalized intersections. An example illustration is shown in Figure 42. Traffic flow was modeled using the cell transmission model (CTM), a micro-to-mesoscopic method for modeling traffic density using discretized sections of each modeled roadway. While CTM does not perfectly model traffic behavior, it provides a solid means for describing vehicle trajectories in a dynamic environment. In each training scenario, traffic was randomly generated on the roadways in the network, drawn from a normal distribution with mean conditions providing oversaturated demand levels for the intersections.

Furthermore, a characteristic of each scenario was that the distributions for the frontage roads had higher mean volume than the distributions of the side street.

Every five simulated minutes of traffic, an updated state description is provided to the agent, detailing each link's CTM density adjacent to the intersection, and the agent is then afforded an opportunity to modify its control scheme.[18] The actuators available to the agent control what portion of each five-minute increment is dedicated to serving traffic from the frontage roads versus traffic on the cross street, with no constraints on the time allocations. While these scenarios do not encompass the full spectrum of traffic conditions, and do not provide a realistic set of policy constraints, the experiments conducted on these scenarios nonetheless provide strong takeaways as to further design considerations, which will be discussed later.

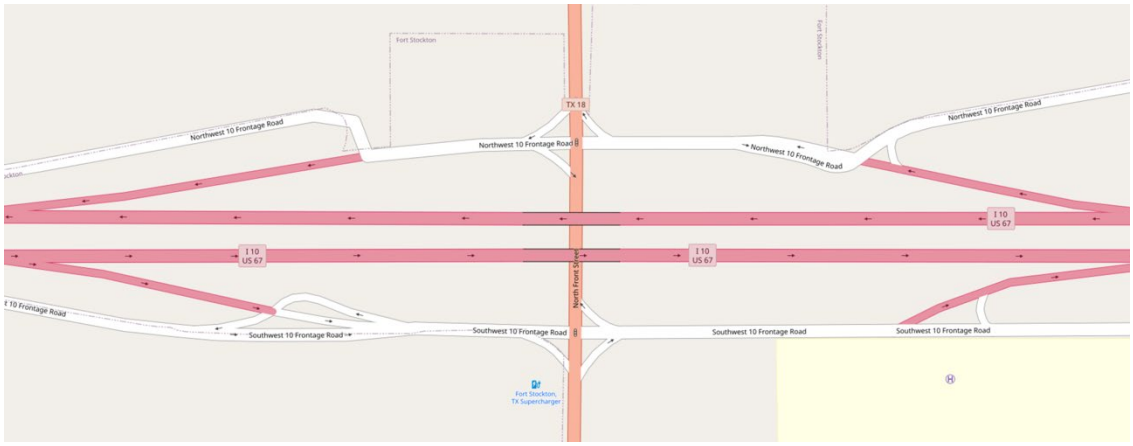


Figure 42: Example Scenario Illustration.

9.2.2. Training Environment Development

To avail ourselves of common and well-researched RL algorithms, an interface was developed that allows our experiments to be conducted in the OpenAI framework. By adapting our scenarios to fit this interface, we can utilize prepackaged and validated algorithms, known as the stable baselines, in training an RL agent. Since our state space representation (i.e., the roadway densities) and the actuators available to the agent are both continuous in nature, we selected the soft actor-critic (SAC) learning algorithm as our performance and learning algorithm.

For our evaluation standard, we selected total intersection delay to measure the agent's performance. While such a metric would not be terribly practicable in real-world trials, it serves as a good intuitive goal for our agent to minimize the amount of delay it causes. In the next

[18] Density on a roadway uniquely describes the traffic conditions present, so this provides an excellent representation of a given link's congestion level.

section, we discuss why this goal, while superficially agreeable, induces a policy that is far from desirable in practice.

9.2.3. Results and Discussion

Training of the initial RL agent resulted in the development of a consistent and definitive strategy that the agent deemed optimal given its reward function—to simply allocate all time to the (higher-volume) frontage roads and none to the cross street. The reasoning behind this is easy to understand: given that all the agent cares about is total delay, it sees no reason to provide preferential treatment to the vehicles on the side street. The agent can minimize delay by simply focusing on the roadway with higher volume—the demonstrable unfairness of never serving the crossing traffic is lost on the agent.

Results clearly illustrate the criticality of care in designing a reward mechanism. While an evaluation function such as total delay may seem, at a high level, like an obvious choice, the repercussions of such a policy are often severe and highly undesirable. This is due to the evidence that an RL agent will, after sufficient training, develop a strategy to perform its task exceedingly well, subject to whatever may define performing the task "well." Equity considerations are necessary in the design of proper reward functions.

In the next section, we discuss a variety of improvements that will allow for more realistic TSC interaction, traffic flows, and sensor data. These steps create a more realistic and, thus, more challenging training environment, in which more care will be taken in creating a performance metric to alleviate the non-egalitarian difficulties imposed by a pure reward-driven agent.

9.3. Further RL Modeling

Despite the core takeaways of testing in the toy intersection, it does not reflect a realistic application in the field. In this section, we detail a more appropriate testbed for training an RL agent in traffic management, one that better reflects the conditions and resources available in a field experiment. As a proof-of-concept demonstration of the proposed RL methods, we developed an experiment in which an intersection was modeled with four approach directions, each having the movement of vehicles modeled in the PTV VISSIM traffic microsimulation application. The approaches were of unequal demand, with two directions (the "main road") featuring volumes of 2,200 vehicles per hour and the remaining directions (the "side street") featuring a volume of 800 vehicles per hour. Traffic on the main road was divided roughly into 60 percent through-traffic and 20 percent each for left and right turns. Side-street traffic was divided evenly among through, left-turning, and right-turning traffic, with no slip lanes through which right-turning traffic could avoid the impacts of the simulated traffic signal. Pedestrian phases were modeled as running concurrently with the through movement of the parallel

roadway. Minimum phase recall was active during the experiments for all approaches, and pedestrian recall was active for all through movement phases. Figure 43 provides an illustration of the test intersection.

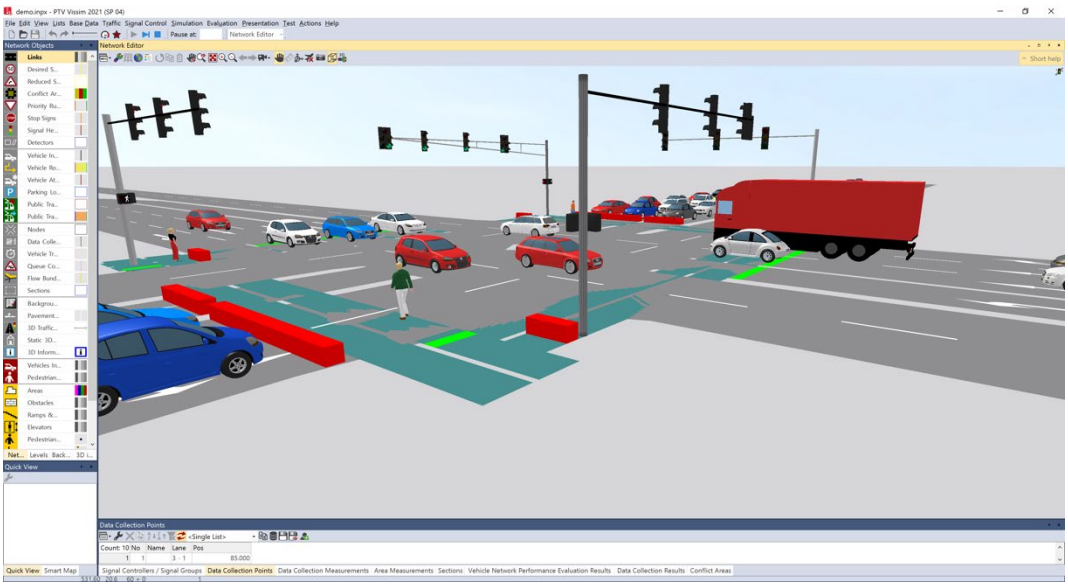


Figure 43: PTV VISSIM Microsimulation Software.

The signal simulation at this intersection was handled by an emulated, NTCIP-compliant Econolite ASC/3-2100 model traffic signal controller, with simulation time divided into green-time allotments for the main road and side street, as well as split settings for left turns vs. opposing through movements.[19] This reflects a simple ring-barrier control schema with minimum timings for any single phase, including pedestrian timing intervals. To develop a suitable initial condition on the roadway (i.e., one in which the agent does not start controlling the network when the roads are devoid of cars), an initial warmup period was conducted in which the signal control policy was held constant.

This configuration was used for the first fifteen minutes of the simulation before control of the signal timing policy was turned over to the RL agent. A full hour of traffic conditions was simulated, yielding forty-five minutes in which the agent could train and modify the TSC configuration. Each simulated hour was divided into five-minute periods or timesteps, after which the agent would receive state information from the environment and, in turn, make adjustments to the TSC configuration. The overall length of the simulation, the length of the warmup period, and the resolution of the timesteps (i.e., the amount of time each timestep lasts) are adjustable, and future work may investigate the impacts of changes to these variables.

[19] Datasheet: http://www.econolite.com/wp-content/uploads/sites/9/2018/10/LEGACY-controller-asc3-datasheet.pdf

9.3.1. State evaluation

To simulate data that can be collected from sensors along a corridor, VISSIM's data collection measurement functionality was incorporated into the gym environment. Data collection points (the simulated equivalent of field sensors) were placed approximately 350 feet upstream of the stop bar, collecting data from each lane of the intersection (left turn bays excluded). From these points, aggregated data was calculated in VISSIM for each approach's volume, average arithmetic speed, and average harmonic speed within the last five-minute simulation period. This data was retrieved through VISSIM's common object model (COM) application programming interface (API). In addition, the agent received information from the virtual traffic signal controller reflecting its current cycle length and split configurations. These values constitute the agent's state space, in which the agent explores which policy options perform best in maximizing its reward function. However, our gym environment has been developed with extensibility in mind. Any number of sensors can be developed that can provide information on the state of traffic at the intersection by following a lightweight and straightforward interface. It should be noted that, as the state space grows in dimensionality, learning for the agent will slow due to an inundation of data and an increase in combinations of state data for which the agent may explore action policies.

9.3.2. Action choice

Three varieties of controls (actuators) were developed for the RL agent: a cycle length actuator, a barrier-split actuator, and a phase-split actuator. For the first of these, the agent can choose the amount of time between the start of a given cycle (i.e., when a particular phase turns green) and the start of the next cycle (when the same phase turns green again after serving all other phases). This actuator is bounded on the lower end by the minimum amount of time that can be served without disrupting any of the phase's minimum timing requirements, including the minimum amount of green, yellow, and all-red time, plus pedestrian walk and clearance intervals, if applicable. Minimum values are fixed at the start of the simulation and must be held constant during training to preserve a fixed action space for the cycle length actuator. Additionally, the programmatic upper bound on the cycle length for the Econolite ASC/3-2100 controller is 255 seconds.[20]

[20] Note that this is not a constraint of the experiment itself and can be relaxed using other controller software.

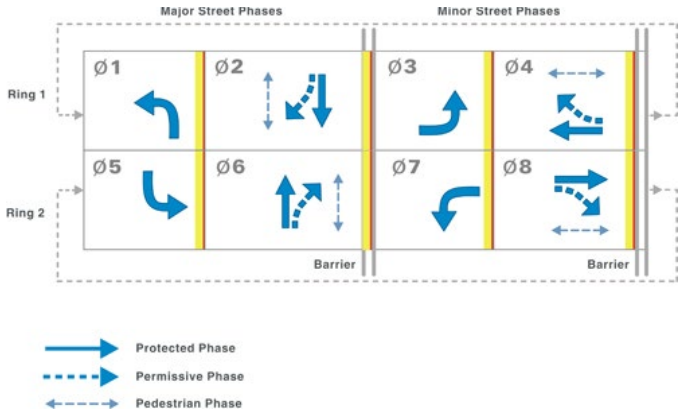


Figure 44. Ring-barrier timing plan visualization (credit: Federal Highway Administration).

The barrier-split actuator defines the amount of cycle time served to the major street vs. the minor street, as illustrated in Figure 44. In practice, this actuator is implemented by providing the agent a choice of percentages of time that will be allocated to the major street, in increments of 2 percent. The amount of time allocated by this actuator is that which is in addition to the required minimum timings for each street's phases. In initial experiments, the minimum time for each street was thirty seconds, so the cycle length actuator was bounded on the lower end at sixty seconds. If this was the value chosen by the cycle length actuator, the barrier-split actuator would have no impact, as there is no additional time that can be allocated to either street. For all other values, the agent can select to allocate 0 percent, 2 percent, 4 percent, etc., up to 100 percent of the excess cycle time to the main street, and the remainder is added to the side street. Rounding errors are handled by rounding down any fractional seconds, then allocating any leftover time randomly between the two streets.

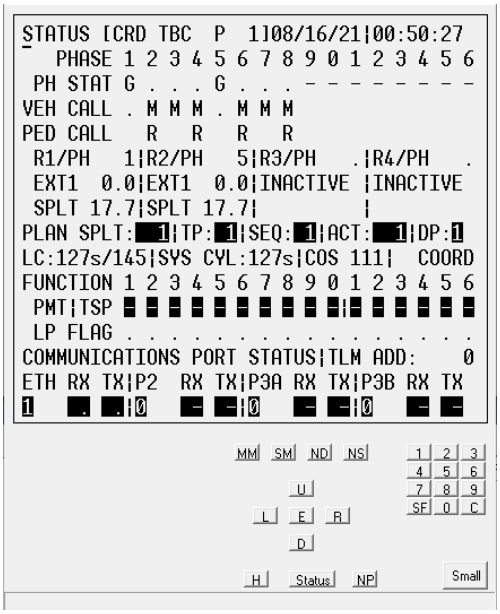


Figure 45. Econolite ASC/3-2100 virtual traffic signal controller

The phase-split actuators behave similarly to the barrier-split actuator. A "subring" is a grouping of phases in a timing plan which share a common ring and barrier. In the timing plan illustrated in Figure 44, as well as in our experiments, there are four subrings in the timing plan. For each subring, the minimum time of the subring is calculated at the start of the simulation, and at each timestep the minimum value is subtracted from the amount of time allocated by the cycle length and barrier-split actuators. The remaining time is to be allocated between the phases of the subring, and, just as with the barrier-split, the agent chooses a percentage (in increments of 2 percent) of the excess time to be allocated to each phase. Rounding is handled identically to the barrier-split actuator. To some extent, the phase-split actuators can be seen as adaptations of the barrier-split actuator, however the barrier-split actuator seeks to split time between two streets, the phase-split actuator seeks to allocate time between, for example, a through movement and its opposing left turn phase.

It should be noted that no two intersections are exactly alike, and many TSCs feature timing plans that deviate from the traditional eight-phase ring-barrier timing plan format illustrated in Figure 45. To that end, we have designed these actuators to be as extensible and accommodating of signal timing plan variations as possible. As such, the number of phases, their order, and the number of streets that can be controlled by this agent are limited only by the practical requirements for their scenario, either by the fundamentals of traffic signal control, or by the implementing TSC.

9.3.3. Performance

As discussed previously, it is evident from initial testing that a well-designed performance metric is essential to achieving good performance for an RL agent, especially in a traffic control scenario. In this phase of testing, we first developed a critic that reflects the critic used in prior testing—overall delay induced by the TSC. To achieve this, at the end of each time interval, the agent retrieves VISSIM-collected measurements of the number of vehicles v and the average per-vehicle delay d encumbered by vehicles along each approach during the interval. These values are multiplied to get the amount of delay incurred during the interval on each approach a, and the negated sum is provided as the reward[21] D for the RL agent for that timestep:

$$D = -\sum_{a} d_a \cdot v_a$$

[21] Recall that, since the agent wishes to *maximize* its reward, a well-designed environment will provide a reward that is *more negative* as the performance degrades. Therefore, the higher delay induced by a poorly performing agent must be more negative, so the negation of the total delay is selected as the reward function rather than simply the overall delay.

Upon completing implementation of this critical functionality, testing was performed to validate that the agent was, in fact, learning to perform better over time. A randomized sample ($n = 30$) of episodes' TSC delays was drawn in which the controller was operated under four timing plan configurations:

- an untrained agent which had not learned at all controls the timing plan
- an agent controls the timing plan after experiencing 500 simulation hours[22] of training[23]
- a fixed timing policy chosen at random which includes a ninety-second cycle length, and
- a fixed timing policy designed in accordance with the ubiquitous Webster minimum-delay formula, $C_L = \frac{1.5L+5}{1-\sum_i \frac{v_i}{s_i}}$ (Webster, 1958), which yielded a 78-second cycle when given the expected hourly roadway volumes

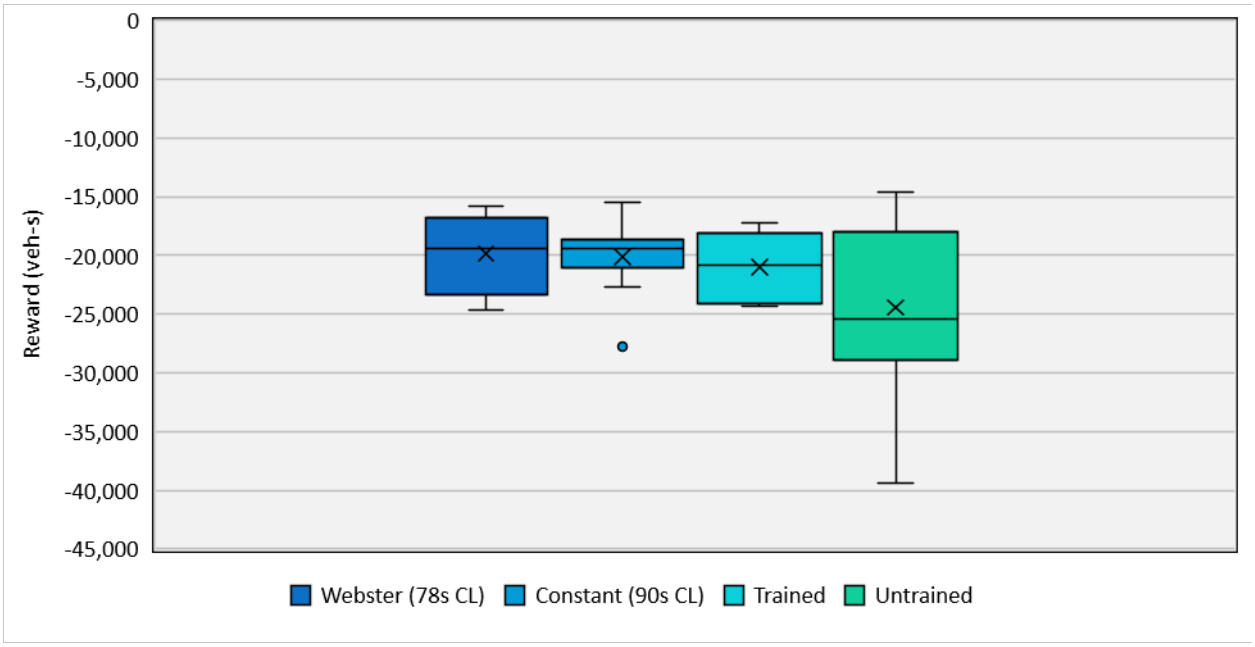


Figure 46: Reward distribution boxplots for four timing policies.

As illustrated in the boxplot in Figure 46, and as should be expected, the untrained reinforcement learning agent performed significantly worse on average than the other three designs. A comparison example of the pre- and post-training cycle length selection behavior is shown in Figure 47. Note that, after training, a narrow band of cycle lengths has emerged that

[22] Note that this amount of training, yielding 6,000 training steps, is not considered extensive, and is chosen to reflect an approximation of what could be computed in an average calendar day. This test is designed to measure learning *momentum* rather than fully learned behavior.

[23] Upon terminating agent training, the *policy* is held fixed, i.e., given an identical set of state observations (data from sensors), an identical timing plan should result for the ensuing time interval. The timing plan can, however, change based on the most recent sensor data.

the agent favors. This trained agent, while not matching the constant-time policy or the Webster-based policy in performance, shows promise when compared to the untrained agent and utilizes policies that are quite close to the chosen constant cycle length. As a result, it can be concluded that the agent is learning better policies to improve its performance. Furthermore, it should be noted that this relatively low amount of training yields results that are, while not as good as those from Webster's formula, are competitive in nature and provide hope for an agent that can outperform traditional techniques.

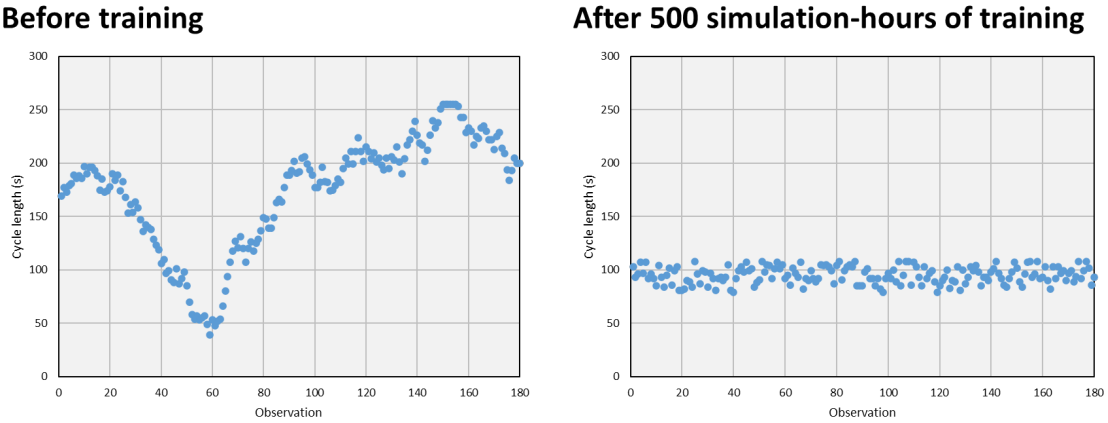


Figure 47. Comparison of selected cycle lengths during fifteen episodes before and after training.

9.3.4. Extended testing

In the previous section, we adapted a VISSIM environment to meet the interoperation needs of an RL agent and performed limited training to validate the agent was able to learn a policy that competes with standard fixed timing models. We now turn our attention to a larger challenge—responding to changes in traffic conditions due to an incident. After confirming the functionality of the constructed gym environment, the environment was extended to present a more significant challenge to the RL agent. This included extending the roadway network to include a model of a freeway segment parallel to the major road. On- and off-ramps were added to connect the freeway links to the major road, similar to the design of a frontage road-based interchange.[24] The test network is illustrated in Figure 48.

[24] One key difference between the experiment design and a standard frontage road interchange is that the at-grade intersection as well as entrance and exit ramps are located in the median in this simulation, rather than on the outside shoulder. This design was chosen to simplify the design of NTCIP-compliant actuators.

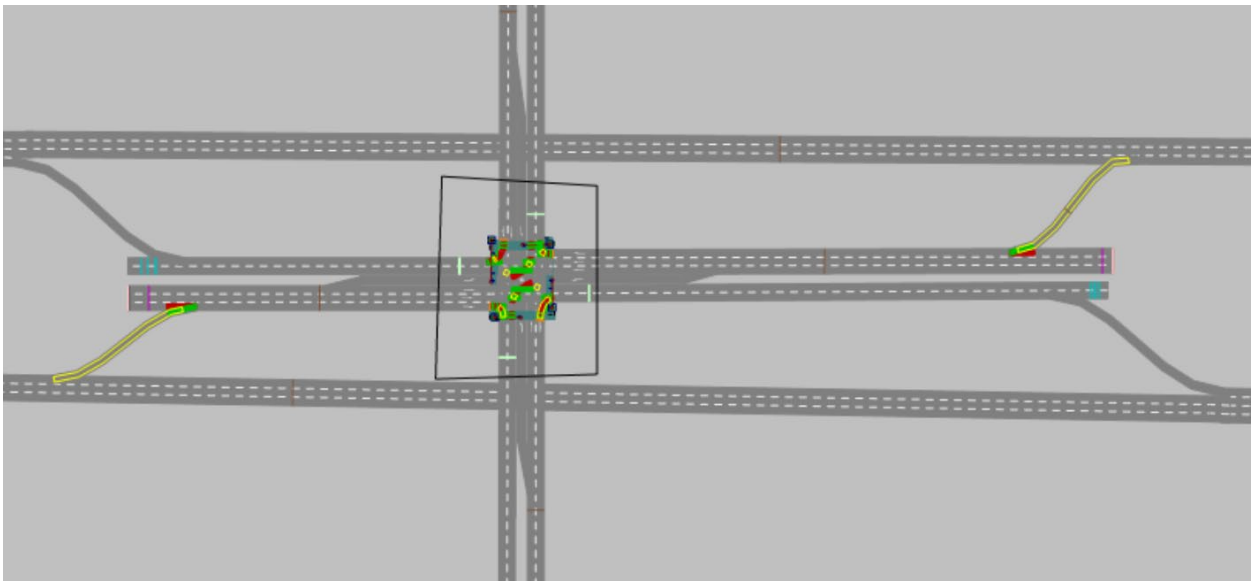


Figure 48. VISSIM network model with freeway and "frontage road" in median intersecting a cross street.

For this experiment, stochastically generated hourly traffic volumes for the freeway links average 6,500 vehicles per hour per direction, and the "frontage roads" each have an additional 400 vehicles per hour in each direction. The cross-street traffic volume averages 600 vehicles per hour per direction. On average, 7.5 percent of stochastic freeway traffic exits to the frontage road in each direction. Of that volume, 25 percent turns left at the traffic signal, 25 percent turns right, and the remaining half continues on the frontage road. For cross-street traffic, 20 percent turns left, 20 percent turns right, and 60 percent continues straight on the cross-street.

As a further challenge to our agent, 15 percent of all simulated scenarios (in training as well as evaluation) generate a complete blockage of one direction of the freeway links, diverting all traffic to the frontage road for approximately twenty minutes. During this time period, the volume of traffic on the frontage roads causes the traffic signal to be well over capacity in one direction. The blockage begins five minutes after the agent takes control of the timing plan.[25] The goal of this challenge is for the agent to recognize, based on its sensor data, that there is anomalous demand, and to respond by altering the signal timing plan accordingly.

Given the extended use case in this experiment, the sensor set available to the agent was extended to include sensor data for the freeway links as well as the exit ramps from the freeway. The additional sensors detect the state of freeway traffic downstream of any blockage, so the effect of such a blockage would be a dramatic drop in volumes and a lack of

[25] Due to the Markovian nature (sometimes called the "memoryless" property) of our RL environment, there is no impact of time on policy choice. Therefore, the agent cannot "predict" the arrival of a blockage.

available speed data. On the other hand, the upstream ramp would reach capacity quickly and speeds would then drop.

9.3.4.1. Performance

To avoid the unintended behavior witnessed in earlier experiments, the performance metric was adapted to include an equity component. While it is still desirable to minimize delay, the performance of a traffic signal should also attempt to distribute the delay it creates in a way that is fair to drivers from all directions of travel. To do this, we incorporate a linear scaling coefficient to our critic function:

$$R = D(1 + G_D)$$

The coefficient G_D is a measure of statistical dispersion of delays amongst the vehicles traveling through the intersection. This approach is based on the Gini coefficient (Gini, 1921), a measure of income or wealth inequality that ranges from zero to one, in which a value of 1.00 represents complete inequality in a population of n (i.e., one person having 100 percent of the wealth). For our purposes, we adapt the notion to measure the dispersion of delay d incurred by each vehicle. The metric is defined as:

$$G_D = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |d_i - d_j|}{2n\bar{d}}$$

With this modification to our reward function R, we increase the penalty when delay is distributed unevenly, but it remains the same if delay is dispersed equitably.

The modified agent was trained for approximately 80,000 simulation steps, totaling approximately 7,200 simulated training hours. This has led to significant performance in the reward collected, as illustrated in Figure 49. A randomized sample ($n = 30$) of traffic conditions for a simulated hour were used to evaluate our four timing policies in scenarios in which no blockage occurred, and another randomized sample was drawn for scenarios in which the freeway was blocked. We again compare against two baselines: a randomly selected policy with a 240-second cycle length, and a timing plan devised in accordance with Webster's formula, with cycle length of 109 seconds, and allocating splits based on relative volume. Here, we see the trained agent performs significantly better than the two baselines and the untrained agent. We also see this in the overall delay accumulated, as shown in Figure 50. However, this is not the only improvement we see in the agent's performance, as it also outperforms all other tested timing policies in terms of equity, as illustrated in the boxplot in Figure 51.

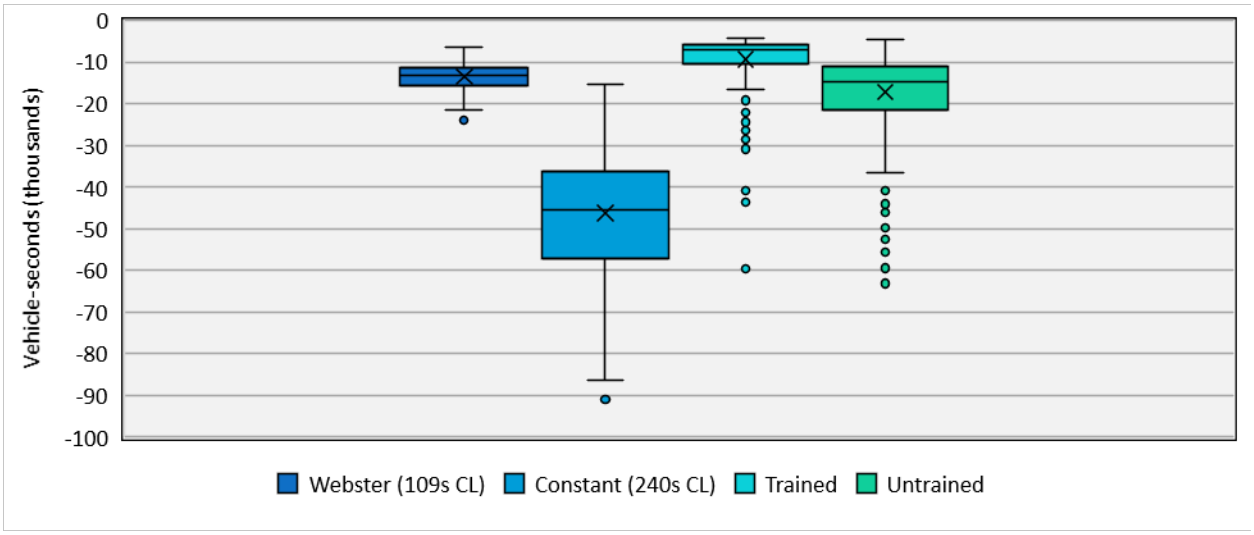


Figure 49. Reward collected in simulations without a blockage.

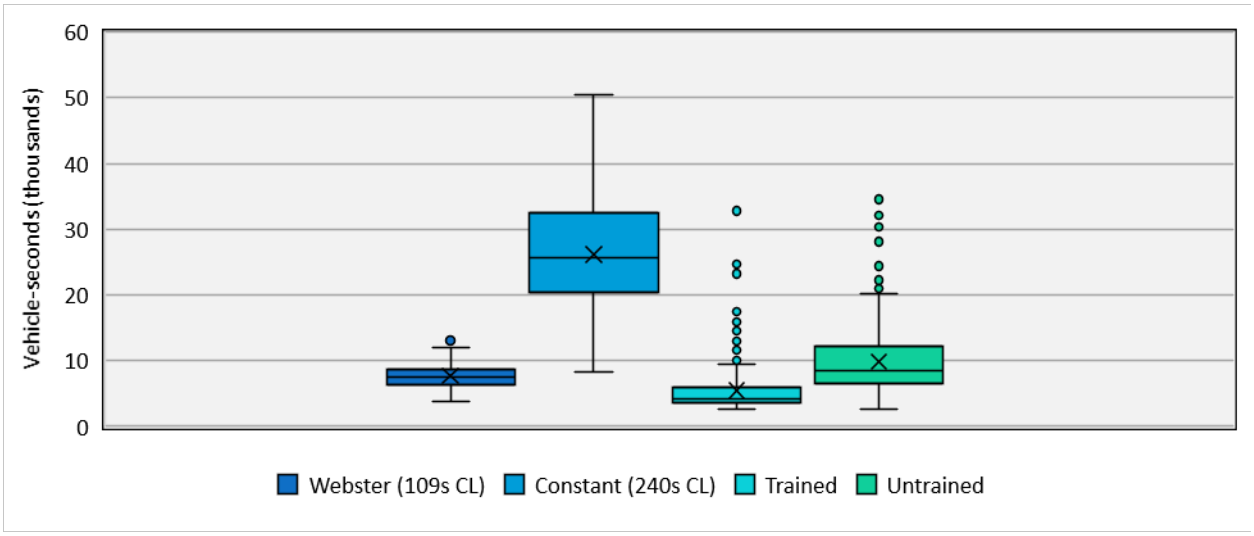


Figure 50. Delay accumulated in simulations without a blockage.

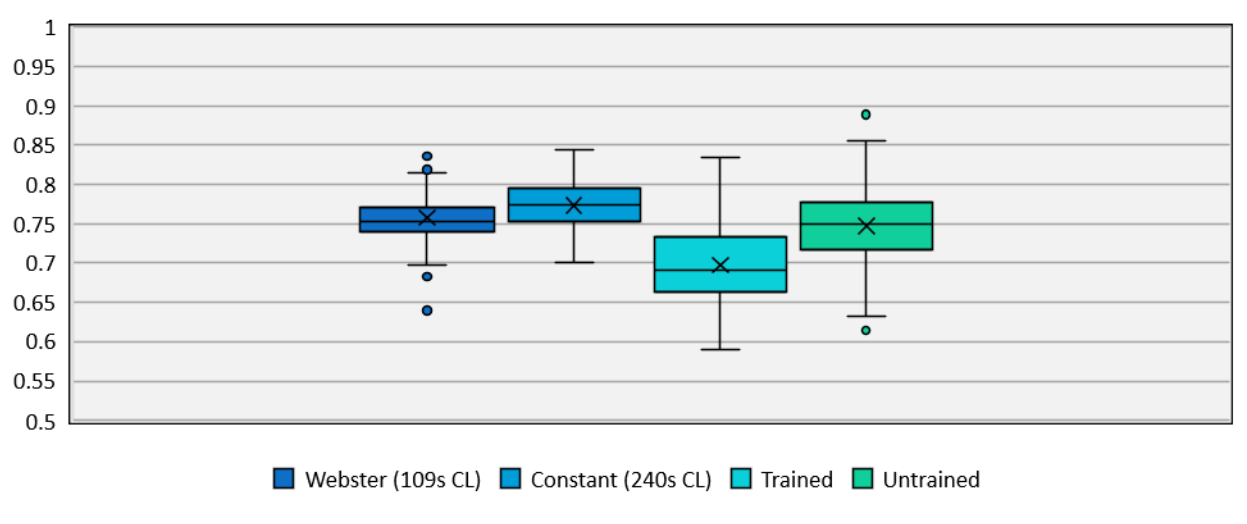


Figure 51. Equity (Gini) coefficient for simulations without a blockage (lower is better).

We further compare the two agents and baselines in the scenario in which a blockage has occurred on the freeway link, leading to oversaturation on one approach. As can be seen in Figure 52, the trained RL agent policy once again is the best performing of the group, although this time in a much more tailored fashion. While the median scenario is handled better than the Webster-based timing plan, on average the Webster-based timing plan reduces the delay, illustrated in Figure 53, by a margin of approximately 100 vehicle-seconds. The equity of the trained agent is also slightly improved compared to the Webster-policy, as shown in Figure 54.

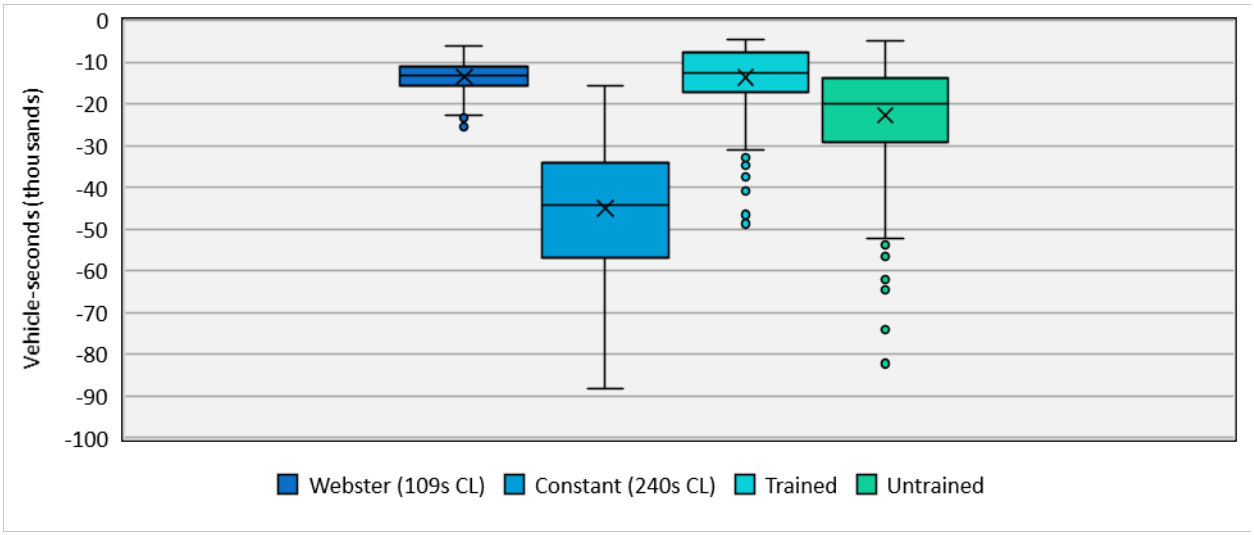


Figure 52. Reward collected in simulations with a blockage.

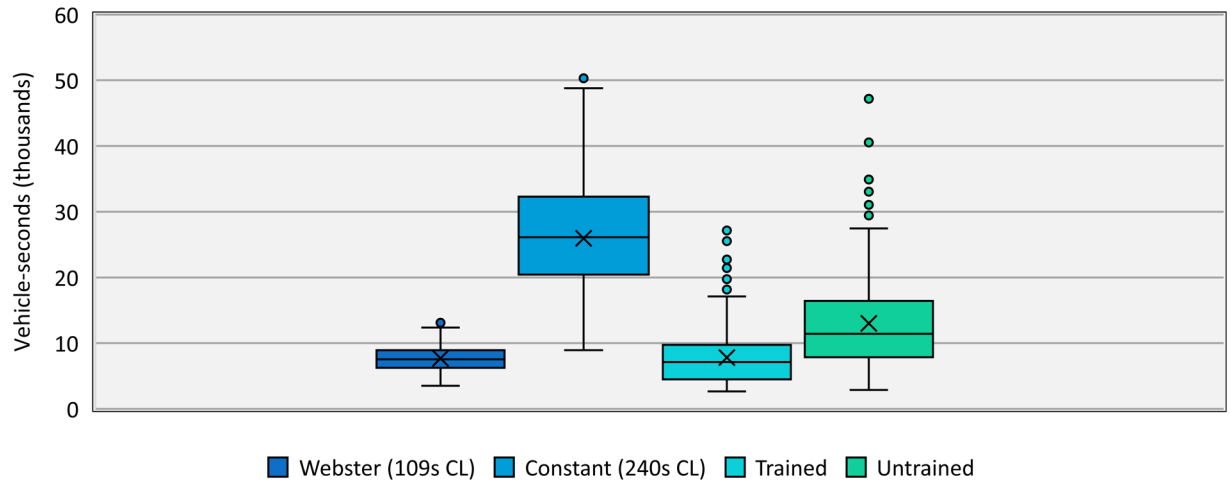


Figure 53. Delay accumulated during scenarios with a blockage.

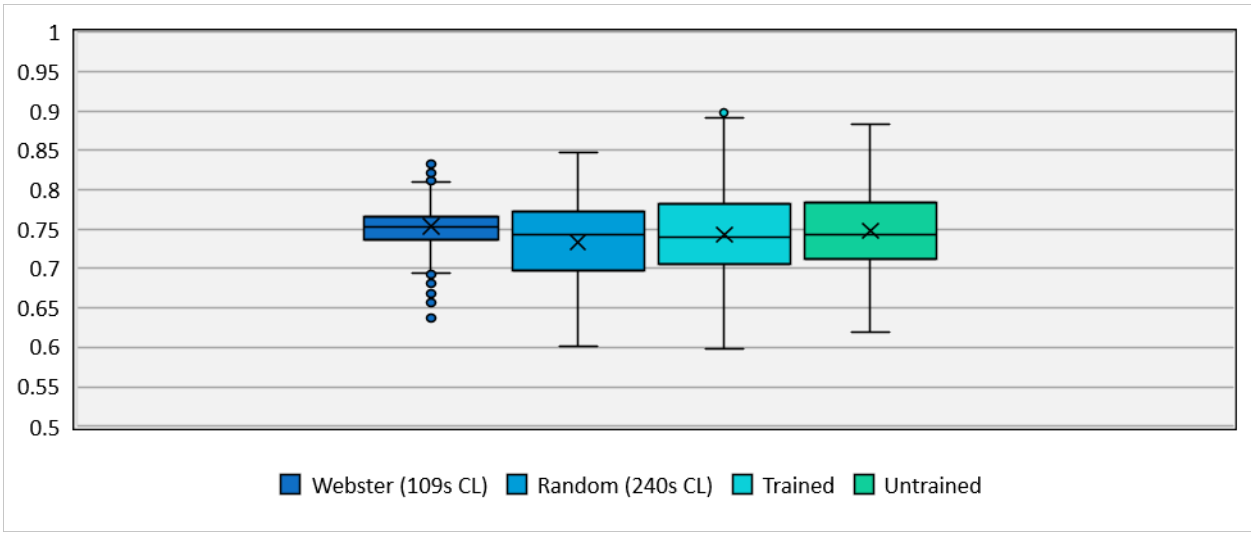


Figure 54. Equity (Gini) coefficient for simulations with a blockage (lower is better).

9.4. Conclusions

To demonstrate the applications of reinforcement learning in an integrated corridor management scenario, we first showed that an essential aspect of designing such an application is the reward function. In initial experiments, an agent with wide liberty to decide signal timing policy and a focus solely on minimizing delay achieved this goal by creating timing plans that were distinctly unfair to users of a lower-volume street. Thus, a modified reward function would be developed to account for this. In addition, the interface through which the RL agent interacted with the initial experiment environment is decidedly not standards compliant. To address this, a new action interface which complies with NTCIP standards became necessary.

In building such an interface, a new traffic simulation environment was required to replace the initial cell transmission model for traffic propagation. The PTV VISSIM micro-simulation package not only offers emulation capabilities for several modern traffic signal controller models, but also provides more accurate vehicle dynamics and more powerful and realistic tools in measuring the performance of the network. We are able through VISSIM to more accurately replicate the real-world sensor data that would be available to a field implementation of this RL model. In these experiments, we see that training an RL agent to manage a TSC is a significant task, one which will take hundreds of simulated training hours to compete with fixed signal timing plans.

However, the key advantage of an RL traffic management agent is the ability to respond in near-real time to anomalies and variations in roadway conditions. To illustrate this, we extended our roadway network to include a simulated freeway on which blockages may occur and which will drive significantly higher traffic through the RL agent's intersection.

We see from the results that the trained agent is superior in terms of equity, delay, and combined reward in an everyday scenario, and is competitive with the Webster-based timing plan when facing a large obstruction. We also see, however, that in instances in which the agent must respond to a blocked roadway, there is a significant increase in the variance of all metrics.

From this, we observe that while an RL agent may take significant training to compete with fixed-time policies, its ability to rapidly respond to changing conditions without human intervention is an asset that warrants further testing in the field. Alhough the increased variance from the trained agent is notable, it has yet to be determined if the variance can be reduced with further training. Nonetheless, the agent has effectively proved to be a suitable agent for managing traffic signal timing plans in simulation, and its practical effects in the field pose an exciting area for future study.

9.5. Pathway to Implementation

This document outlines a series of considerations and steps that should be taken to conduct a field experiment that trains a reinforcement learning agent to automatically detect traffic conditions and respond to them by adjusting an intersection's traffic signal timing plan. This experiment will perform initial training for the agent in a simulated environment that closely reflects the chosen testbed location. The agent will learn from the data it obtains from real-world sensors and adapt a signal controller to respond to its observed traffic conditions. The agent will train to maximize a predetermined reward function that characterizes its performance based on traffic conditions that result from the agent's choices and external factors such as human behavior.

The goal of this experiment is to develop an agent that can respond intelligently to changes in traffic conditions and provide suitable signal timing configurations without human intervention. To accomplish this, several preparations must be made before the experiment can begin, and we document these herein. A field implementation pathway, illustrated in Figure 55, can be broken into independent pieces, each requiring distinct data. The first phase entails modeling the geometry and behavior of traffic at the selected test intersection, followed by adapting the available sensor outputs to meet the needs of the learning agent. We then simulate these characteristics and outputs to train the agent offline to give the agent some degree of experience prior to implementation in the field. Once the agent has been connected to field equipment and allowed control of the traffic signal controller, its performance will then be evaluated to confirm proper operation in line with prior simulated behavior.

Figure 55. Field implementation pathway.

9.5.1. Site Location

Given the relatively high frequency with which they are implemented in the field, an ideal testbed for our reinforcement learning agent would be a four-way, eight-phase intersection that provides four through-movements and four left-turn movements. The traffic signal would preferably be one in which a standard ring-barrier timing pattern is the default, with no phase reservice or other atypical timing plan features. The virtualized experiment showed success in operating such a traffic signal, including the pedestrian phases; however, these pedestrian phases, if included, would ideally be non-actuated, as current design does not include a mechanism for handling adjustments to phase length caused by pedestrian phase calls.

An ideal testbed location would also be sufficiently distant from other signals, so as to reduce the influence of one signal's timing plan on another's performance. While this is not a requirement for implementation testing, should a poorly performing signal timing plan be implemented by the learning agent, a risk of causing queue spillbacks through nearby intersections is present. Therefore, isolating the testbed will provide a means to reduce this risk. In further experiments, this constraint may be relaxed in order to train a reinforcement learning agent to adapt coordinated corridor timing plans.

9.5.2. Hardware

9.5.2.1. Traffic sensing

For a quality implementation of this experiment, sensors must be deployed in the vicinity of the test site to measure both the state of traffic to which the reinforcement learning agent will respond, as well as the agent's performance in responding to its observed states. In measuring the state of traffic, simulated training depends on vehicle counts and average speeds (both harmonic and arithmetic) to define the scenario on which the agent should base its policy choices. To replicate this in the field, vehicle counters that can provide real-time (or near-real-time) data should be made available, and a module should be provided that describes the average vehicle speed on approach roadways. Possible choices include radar-based sensors, like Wavetronix or camera-based techniques such as GRIDSMART, to provide volume data. INRIX may be a good source for speed on approaches. This data will be used in deployment to reflect the state of traffic in which the agent is operating and will mirror the simulated data used in initial training.

Additionally, the agent must also be able to receive sensed feedback that evaluates its performance. In initial experiments, the agent performance was measured according to the amount of delay incurred per vehicle. In later stages, this was expanded to include an equity metric that accounted for how equally the agent treated each direction of traffic. This modification was necessitated by the agent's reaction to the perverse incentive of minimizing total delay—when demand is present from each direction, the delay-minimizing strategy simply serves as much green time to the busier roadway while ignoring the other.

To recreate this measure in the field, it is unlikely that per-vehicle delays can be reliably obtained. However, if an average delay value can be obtained and volume data is readily available for each movement, an approximation of such a method becomes available. INRIX is currently testing signal performance measurement tools that include data on volume and delays per turning movement. This may be sufficient to replicate a measurement of performance for the agent that was previously trained in a high-resolution simulator that provides per-vehicle data. Requirements for communication of sensor data to the management engine are not yet defined and will depend on the hardware implementation chosen.

9.5.2.2. Traffic signal control

For this experiment, a traffic signal controller that is capable of NTCIP communication is required, specifically one that can send and receive SNMP (the protocol underlying NTCIP) packets via a networking connection. Such a network connection need not be of excessively high speed, but a minimum performance metric would be 1 Mbps up/down, and higher-speed

connections would come with faster performance. To facilitate ancillary functions of the traffic signal controller such as status and error reporting, a network switch may be necessary to provide communications not only to the experiment management computer but also to a supervising traffic management center or other facility.

9.5.2.3. Experiment Implementation

As discussed in the next section, the experiment requires the ability to simulate the test site intersection in a reasonably realistic microsimulator such as PTV VISSIM. To do so, a computer must be available that can perform these simulations quickly. Initial experiment testing used an Intel Xeon four-core processor running at a clock rate of 3.6 GHz, with eight logical processors and 64 GiB of physical memory.

The storage space required for this experiment is not particularly high, as, excluding installed software and operating system usage, project resources require less than 256 MiB of on-disk storage. Therefore, a low-cost storage device with 256 GiB or more would be sufficient for the test computer. However, given the performance impact, a solid-state drive would be preferred (but not required) over a spinning-disk hard drive.

9.5.3. Software

9.5.3.1. Traffic signal control

The traffic signal controller that implements this experiment must feature NTCIP support, specifically compatibility with the NTCIP 1202 specification. While initial testing was conducted with a simulated Econolite ASC/3-2100, any controller capable of implementing these features should suffice. As of this writing, testing is incomplete to determine which version of NTCIP 1202 is required for compatibility with the learning agent as configured, but it is the general belief that 1202v02 support should be sufficient.

To further limit potential compatibility conflicts, it is recommended that the selected traffic signal controller be available in simulated form within the microsimulator used for offline training, and a basic configuration of a signal timing plan should be provided both in the simulated and field controllers upon which the reinforcement learning agent can train and adapt. This includes configuration details such as phase minimum recalls, detector locking status, and others. By only exploring a limited subset of the parameters of a preconfigured timing plan, learning will proceed at a faster and more reliable pace.

9.5.3.2. Reinforcement learning agent

To develop test policies for the learning agent, a simulated environment should be established in a microsimulator. In initial-phase testing, this was completed in PTV VISSIM, version 2021. The microsimulator provided a simulated traffic signal controller meeting the specifications discussed previously, as well as an interface for obtaining simulated sensor data. As PTV VISSIM is only available for computers running Microsoft Windows operating systems, this forms a constraint if VISSIM is chosen as the preferred simulation software.

While VISSIM provides a set of sensor data that meets the requirements mentioned earlier, the data is provided at a level of reliability that exceeds real-world capabilities. As such, prior to beginning field testing of the reinforcement learning (RL) agent, an investigation should be undertaken to determine the reliability of the chosen field sensor hardware. Once the sensors' reliability is determined, a software module will be developed that postprocesses the simulator-provided sensor data and adds noise to better reflect real-world hardware conditions.

In addition to a microsimulator package, the implemented reinforcement learning framework depends on a Python interpreter (version 3.7 or higher) as well as several external library packages that are not included by default. These packages each serve a distinct purpose and are enumerated below, along with their dependencies:

- gym – provides a framework for reinforcement learning environments
    - Dependencies: scipy, cloudpickle, future, pyglet, Pillow, numpy
- numpy – provides array computing functionality
    - Dependencies: none
- pandas – provides data structure capabilities
    - Dependencies: numpy, pytz, python-dateutil, six
- pysnmp – provides SNMP communication functionality
    - Dependencies: ply, pyasn1, pysmi, pycryptodomex
- pywin32 – provides Common-Object Model communication with Windows software such as PTV VISSIM
    - Dependencies: none
- stable_baselines3 – provides stable implementations of reinforcement learning algorithms
    - Dependencies: cloudpickle, numpy, matplotlib, torch, pandas, gym, typing-extensions

9.5.4. Training

9.5.4.1. Environment

Since a key implementation detail is being able to perform initial RL training without negatively impacting the actual state of traffic with poor policy choices, it is imperative that a simulated model of the testbed intersection be developed. This model should include both roadway (plan) geometry, speed and routing choices, and volume data for the period(s) in which the agent will eventually operate online (i.e., with control of real signal timing plans). As discussed previously, sensor data from the microsimulation software should be adapted from its "omniscient" data quality to reflect potentially noisy measurements from real-world sensors more accurately, providing lower-quality data to the RL agent.

In designing our simulated environment in this way, the simulated environment will more closely reflect the real test environment, providing a sandboxed learning environment in which the agent may perform initial testing that may result in poorly performing timing plans being evaluated. This reduces the likelihood that such timing plans would be selected in later training when the agent is given control over a real-world intersection controller.

9.5.4.2. Scenarios

To properly train the RL agent in simulation, a wide variety of traffic scenarios should be provided by varying the volumes of traffic on each approach and in each turning movement. To do this, a statistical distribution of travel demand should be developed that details the likelihoods of various volumes for the study period(s). From these distributions, a wide variety of traffic profiles can be developed programmatically and used as training scenarios for simulated RL. Based on the input distributions, these scenarios will provide scenarios that are more likely to occur to the simulator more frequently, while still giving examples of rare traffic patterns for which the agent can develop signal plan policies.

9.5.4.3. Learning

The RL agent will require an extensive amount of training in the simulated environment before being allowed to modify a real-world traffic signal timing plan. The scenarios, as detailed above, should cover a wide variety of potential traffic patterns, including accidents, blockages, and special high-volume events, while still providing a strong focus on typical conditions that the agent will encounter more frequently.

As of this writing, it is unknown the amount of learning time required to reduce the likelihood of exceptionally poor policy performance in the field, and there is no guarantee that an agent can avoid attempting such policies in the field as part of the continued exploration of new

policy options. However, to rein in both the learning time and the amount of poorly performing policies the agent can choose from, constraints can be defined for the actions from which the agent can choose.

For example, if a cycle length parameter can range from 30 seconds to 255 seconds, but it is unrealistic to expect good performance outside the range of lengths from 100–160 seconds, the agent can be artificially limited to only choose from policies in the 100–160 second range. Consequently, though, if a better policy could theoretically be found outside this range, it will not be discovered by the agent and will not be deployed.

9.5.5. Experiment Monitoring

9.5.5.1. Error/conflict handling and reporting

Given the experimental nature of the automated timing plan strategy of this proposal, careful attention must be paid to the actual timing plans that are chosen by the agent once it is connected to a real-world signal controller. Not only is it possible that a poorly performing timing plan is chosen, thus causing severe traffic backups, it is also possible that the controller or learning agent may malfunction, leading to even more serious issues such as a flashing-all-red controller configuration.

To avoid this, several strategies are available. Initially, it is recommended that the learning agent be enabled only for a portion of each day. This will allow for optional human monitoring of the experiment in real time without dedicating considerable human resources to the experiment. In pursuing this, the time periods in which the agent is enabled should be rotated to provide the agent opportunities to train on the wide variety of conditions that may arise.

In concert with rotating the agent's active schedule, a fault-monitoring system should be actively monitored to ensure malfunctions are corrected in a timely manner. Most modern signal controller models feature some sort of fault monitor module, and the agent itself provides a basic level of fault reporting as well. These two together will provide a basic reporting mechanism that can be monitored by human supervisors, either in real time or as a responsive strategy.

One further option for ensuring minimal errors or malfunctions is proactive approval of all timing plans to be tested. While this is potentially resource-consuming and may limit the ability of the agent to respond to unusual conditions with customized timing plans, it ensures the safety of the experiment's real-world rollout. This strategy can take two forms: the agent may suggest a timing plan that could be approved or rejected by a supervisor before being loaded

into the signal controller, or the agent may select from a set of timing plans that have been pre-approved as being safe for implementation.

The former method will considerably slow training and may require modifications to the learning agent to handle situations in which the suggested policy is rejected; however, it retains the ability to provide custom timing plans that may not have been considered by human operators. The latter method will constrain the agent considerably in terms of the flexibility it allows in choosing timing plans, but it will both speed up training and provide stronger guarantees against poor performance. Nonetheless, the constraints imposed are somewhat antithetical to the experiment's purpose, as the selection set available to the agent is reduced to plans of human design and does not provide for exploration of untested plans.

9.5.5.2. Performance monitoring

To properly evaluate the performance of the learning agent, good records must be maintained for the inputs and outputs of the agent. This includes the sensor data provided to the learning agent, the NTCIP communications from agent to controller, signal controller warnings or errors, and any identified external data such as weather conditions, special events, or other information that may skew performance. Since this data can be expected to be large in magnitude, a dedicated storage device independent from the agent may be necessary. Independent storage of historical data not only provides a potential performance boost to the learning agent's host device, it also provides a more resilient storage solution, as we will discuss momentarily. Even if an independent storage device is not implemented, the computer hosting the historical data will require significant storage capacity to maintain records during the course of the experiment.

9.5.5.3. Traveler feedback

One additional aspect that may prove valuable in collecting experimental results is the impact on travelers' evaluations of the signal performance. Providing drivers with a means to report complaints regarding the experiment may provide key insights that may not be evident from the more empirical data collected. A potential downside to this includes a skewed sense of the experiment's functionality, as it may be impossible for travelers to know the rotating schedule implemented, and may provide complaints that are, in fact, not caused by the experiment at all. Additionally, given that humans tend to report complaints far more than compliments, it is possible that this feedback mechanism will not record that experiment performs well.

9.5.6. Evaluation

9.5.6.1. Performance metric

As the most important component of a reinforcement learning agent, monitoring changes in the chosen performance metric is imperative. From initial results, a long-term improvement in the reward function should be observed, leveling off once peak performance has been attained. This data will, of course, experience fluctuations due to ever-changing traffic conditions and continued exploration of policy choices by the agent. Additionally, poor performance may be expected in situations in which the agent cannot determine the traffic state accurately, such as when sensors fail, or when conditions change more rapidly than the agent can respond to them. Since, as of this writing, the agent modifies its timing plans once every five minutes, there is a considerable delay window between a change in conditions, such as a wreck, and the system's response to it. Nonetheless, if the agent performs well, it is expected that it will respond to such incidents quickly and will not, to the extent it is capable, allow conditions to remain for extended periods of time. This is, however, subject to externalities such as the response time to clean up an accident located in or adjacent to the intersection.

Additionally, while only one performance metric can be utilized during training, attention should be paid to whether the initial choice was well-suited to the experiment. It remains to be seen whether a per-vehicle or per-movement equity metric is a better choice and including equity as a component in the reward function can increase delay or other key performance indicators such as greenhouse gas emissions. Therefore, after some period of experimentation with a given metric, the choice should be reevaluated to determine if a better choice can be made for future experimentation. This will depend significantly on any observed side-effects.

9.5.6.2. Side effects

It is highly likely that, given a chosen performance metric, the agent will yield a policy that aims to maximize the reward it obtains from that metric. However, this metric will be the only objective the agent considers, and it may not yield exactly intuitive results. As initial testing showed, a metric that focuses on minimizing overall delay can cause the side effect of under-serving a lower-volume street when constant demand exists from each direction, creating lengthy delays for some while others encounter very little delay. This condition motivated the inclusion of an equity component in the performance metric and illustrates the possibility for unexpected consequences of the design of the metric.

Therefore, it is important to monitor the testbed environment for such troublesome incentives. If some unexpected consequence arises that must be accounted for in incentivizing good timing-plan behavior, the performance metric must be adapted, and training must begin anew.

To limit the impact of such side effects, if they are determined to be of high severity, the agent should be taken offline, and a normal signal timing plan should be put into place while simulated training is conducted with the modified metric. Once sufficient training has been conducted in a simulated environment, real-world testing can resume.

9.5.7. Resiliency

To ensure the integrity of this experiment, resilient infrastructure should be provided for the experimental hardware used in training the agent. While many signal control cabinets already feature a backup power supply and redundant network connections, it is important that these also be made available to the agent's host computer. Doing so will allow for continued operation when a partial failure is detected, and a safe shutdown of the experiment can be completed when a larger failure occurs. This will, in turn, protect the integrity of experiment records and ensure that the signal controller does not receive incomplete configuration data from the agent if power is lost or communication is severed.

In addition to resilient power and communications components, this document has detailed several other steps that can be taken to provide a reliable experiment setup. These include providing bounds on the explorable action space for the agent, proactive or reactive fault monitoring and correction, and providing ample initial training in a simulated environment. While these are by no means guaranteed to ensure a high safety or performance level for the experiment, they will reduce the risks posed by the experiment considerably.

9.5.8. Conclusion

As the primary motive in engineering is creating a safe environment, several considerations are needed to ensure this experiment is conducted without severely hampering the safety of the testing environment. While the technology and human effort we detail in this document goes beyond the typical level required for an experiment in traffic signal timing, it is hoped that these will act as suitable guardrails to provide a reliable experiment that can improve conditions without further human intervention. Vigilant monitoring and evaluation of the agent's performance and the conditions that arise in the field are essential to conducting this experiment successfully and safely, and providing the resources needed to establish a suitable testbed will significantly reduce the risks faced throughout the experiment.

Chapter 10. Short-term Travel-time Prediction

The goal of the models presented in this chapter is to estimate the travel time that drivers will experience along a freeway corridor based on the traffic conditions at the time they start their trip on the corridor. This work uses speeds and traffic volumes as indicators of prevalent traffic conditions.

Estimating experienced travel times accurately based on traffic conditions at the departure time requires considering the spatial and temporal variation of traffic conditions along a corridor:

- Spatial variability is typically captured by reporting traffic conditions for sub-sections of the corridor over which such conditions are likely to be homogeneous. Corridor travel time is obtained by adding up the travel time on each corridor sub-section.
- Temporal variability refers to the evolution of traffic conditions as drivers move through the corridor. In order to accurately estimate the travel time to be experienced by travelers entering the first corridor subsection at time *t* it is necessary to forecast what travel times they will encounter at times *t+1,t+2….t+d* (where *d* is the duration of the trip) on downstream segments.

Short-term travel-time prediction (STTTP) on freeway corridors involves forecasting travel times on corridor sub-segments one to two hours into the future at fine-grained temporal resolution (five to 15 minutes). Predictions are often updated every few minutes and are necessary to estimate realistic end-to-end travel times along freeway sections, particularly when some sub-segments in the analyzed section experience congestion. The aggregation of forecasted travel times across segments is often done using a dynamic approach. In such approach the travel time used for each segment corresponds to the arrival time at that segment based on the travel time experienced up to that point.

The following sections describe two studies conducted during this project to understand the potential and value of machine learning (ML) models in short-term travel time prediction. Section 10.1 presents a preliminary analysis conducted on I-35 using one year of traffic speed and volume data. Section 10.2 presents the results of a larger study that considered eight corridors on two different sites and explored different model specifications and the challenges of field deployment in real time. Training and real-time performance is analyzed in depth and the model value is quantified relative to naïve methods. Section 10.5 provides additional information about data pipelines, code and database documentation, and introduces a web-based application deployed in order to share these models with TxDOT.

10.1. Preliminary Model Development

This section introduces and assess the performance of two types of short-term travel-time prediction models, a time series approach, and a recurrent neural network (RNN) method developed for a 23-mile section of I-35 in Austin, Texas.

The models utilize volume data streamed from sensors on the road and probe-based speed data provided by INRIX to estimate corridor travel times in real time, considering forecasts of the evolution of traffic conditions along the analyzed freeway section. The live predictions may be used to display travel time updates on variable message sign (VMS) boards or to support traffic management.

A common approach to corridor travel-time estimation involves dividing the corridor under study into homogenous sub-segments, measuring segment travel time/speeds at small time increments, and adding the measured segment travel times in order to produce a corridor-level travel-time estimate. The estimate produced using such approach (referred to as static travel-time estimation in this work) reflects the total travel time at the time that the trip begins and assumes that traffic conditions do not change as travelers move through the corridor. The travel time experienced by drivers may be very different from the static travel time, particularly during times of day in which traffic conditions evolve rapidly.

In this study, we seek to improve corridor-level travel-time estimation by computing dynamic travel times. Dynamic travel times are computed by considering, for each segment along the corridor, the travel time corresponding to the arrival time at the beginning of the segment. Such travel time may be different from the travel time measured at the beginning of the trip, and it is not known a priori.

To further illustrate the difference between dynamic travel times and static travel times, consider a corridor that is split into five segments. Table 17 illustrates segment-level travel times along the corridor over five timesteps. The travel times are expressed in terms of number of timesteps, which may have any pre-defined duration. The static travel time for a vehicle starting at the beginning of segment 1 at timestep 0 is 3.8 timesteps. This is the sum of the travel times of all the segments at the starting time of the trip. The dynamic travel time is 3.3 timesteps because conditions change as time proceeds and the travel time of a particular segment when the vehicle reaches it may not be what it was when the trip started. In the case shown in Table 17: Example of static and dynamic travel time computation, the vehicle takes 1.5 timesteps to traverse the first segment by which time the travel time of segment 2 would have changed to 0.4 timesteps. The vehicle finishes traversing segment 2 at 1.9 timesteps and segment 3 at 2.6 timesteps at which point the travel time of segment 4 would have become 1.1

timesteps. Similarly, the travel time of the last segment would have become 0.1 when the vehicle reaches it.

Table 17: Example of static and dynamic travel time computation

Timestep	**Segment Travel Time** (in timesteps)				
	Segment 1	**Segment 2**	**Segment 3**	**Segment 4**	**Segment 5**
1	1.5	0.2	0.6	0.9	0.1
2	1.7	0.4	0.7	1	0.2
3	1.5	0.4	0.8	1.1	0.2
4	1.4	0.3	0.6	1	0.1
5	1.2	0.2	0.6	1	0.1
Cumulative Travel Time					
Static	1.5	1.9	2.6	3.7	3.8
Dynamic	1.5	1.7	2.3	3.2	3.3

Travel time for traversing segment based on static and dynamic approaches
Travel time for traversing segment based on static approaches
Travel time for traversing segment based on dynamic approaches

The approach proposed in this study uses machine learning models to make short-term segment-level travel-time forecasts for each segment along the corridor and uses them to estimate corridor-level dynamic travel times. The following sections present the experimental design, introduce the data sources used in this study, describe model development and training, and provide a detailed analysis of model results.

10.1.1. Experiment design

The short-term segment-level travel-time forecasts necessary for approximating the corridor-level dynamic travel times were made using an auto-regressive timeseries (TS) model and a recurrent neural network (RNN) model. The dynamic travel times computed using the forecasts from these models were compared against the static travel time and the ground-truth travel time. We assume the ground-truth travel times to be the dynamic travel times computed using the actual segment-level travel-time data for each timestep. Note that the ground-truth travel times computed in this manner may not be the actual corridor-level travel times because there will be some error introduced due to the discretization of timesteps and, more importantly, the readings from the sensors may not be accurate. In this study, we attempt to keep the errors due to discretization to a minimum by using a small timestep size of five minutes. Identifying and rectifying the errors in the sensor readings is beyond the scope of this project and therefore the speeds recorded by the sensors were all assumed to be perfectly accurate. Based on these considerations, we assume that the ground-truth travel times are the same as the actual travel times.

10.1.2. Data

This study considers data collected on a 23-mile stretch of I-35 through Austin, Texas, in the year 2019 (Figure 56). Travel-time data was obtained from INRIX, while traffic volume data was provided by smart work zone trailers which were deployed throughout the corridor at positions that remained fixed between 2018 and 2020. While the temporal resolution of the data was as fine as one minute, a five-minute aggregation was used to define both the predictors and target variables of the models. The following sections describe both data sources.

Figure 56. INRIX segments and smart work zone trailer location.

10.1.2.1. INRIX Dataset

The INRIX dataset provides speed and travel-time data on pre-defined segments of varying length. The data is collected from a subset of vehicles that share their positions through in-vehicle GPS systems. INRIX uses the shared data to infer vehicle speed from several points within each segment, and further aggregates it to determine segment-level speeds. The northbound stretch of I-35 under study consists of 42 segments, while the southbound direction includes 43 segments. The segments are not of the same length. Figure 57 shows the distribution of segment lengths in each direction. Counts for the northbound and southbound segments are stacked. While INRIX data is provided in real time, TxDOT has access to archived data. For this effort, researchers downloaded one year of data for 2019.

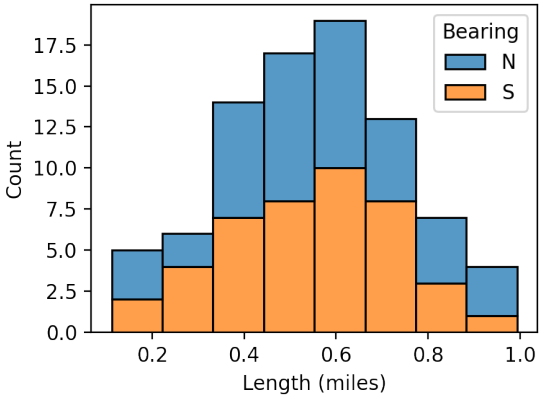


Figure 57. Histogram of segment lengths.

The heatmap in Figure 58 illustrates dynamic travel-time patterns for every five minutes of every day in 2019 where data was available and computed as described in the previous section. Average daily travel times in both directions are 30 minutes (weekdays) and 25 minutes (weekends) in both directions, with standard deviations in the order of ten (weekdays) and five (weekend) minutes. The heatmap clearly shows a morning and peak period in both directions; the northbound (NB) direction exhibits a more pronounced morning peak than the southbound (SB) direction and a severe evening peak period. Morning peak periods in both directions are shorter and more consistent than evening peaks. The banded appearance in Figure 58 is a result of weekends having overall lower travel times, which are reflected in darker bands every five days.

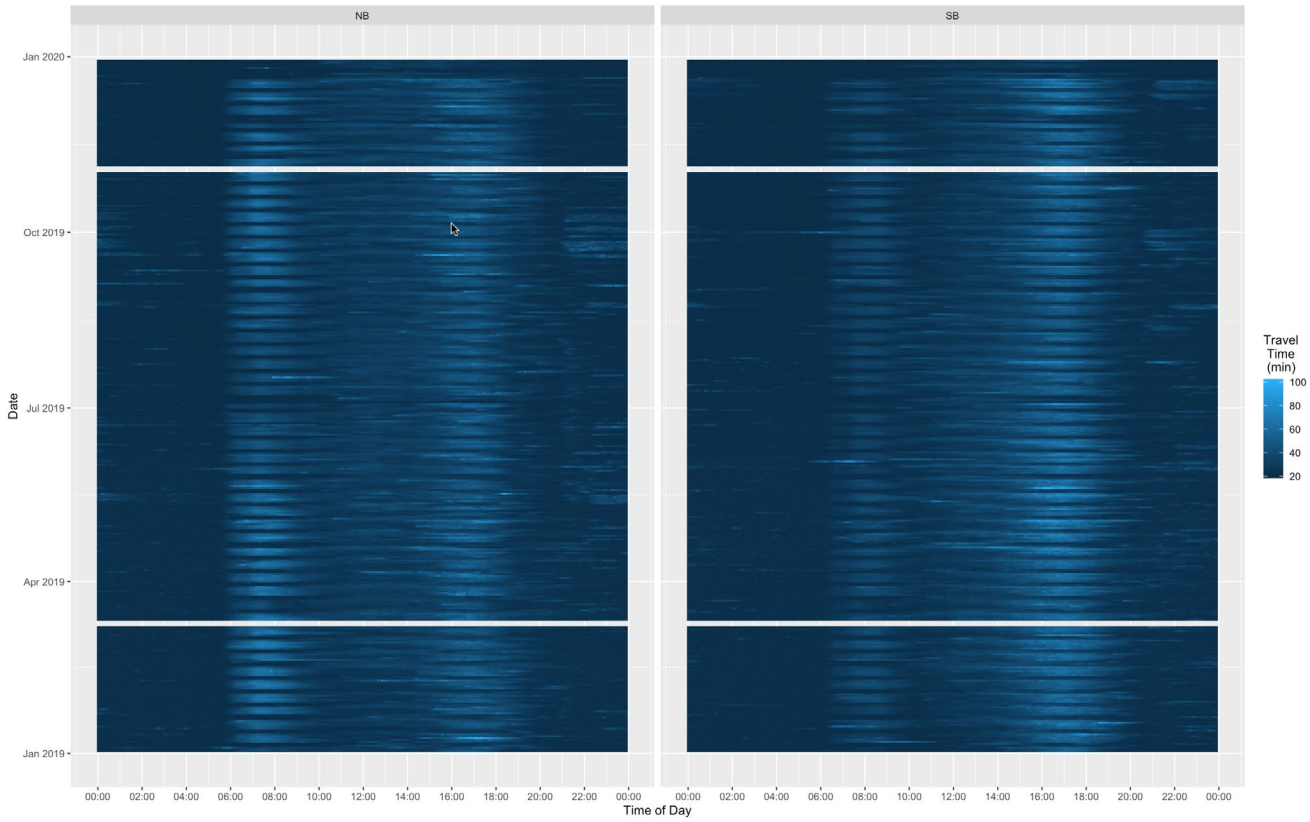


Figure 58. Dynamic travel time patterns in 2019 using INRIX data.

Given that travel times vary by day of week and time of day, we compute average travel times for every 15-minute interval throughout the day using one year of five-minute data.

Figure 59 illustrates such travel times and corresponding standard deviations for one weekend and one weekday. In the context of this study, we consider that the travel time on a particular timestep is “typical” when it falls within one standard deviation from the corresponding mean value (based on time of day and day of week).

Days and times with “atypical” travel times are somewhat noticeable in Figure 58, and are explicitly illustrated in Figure 60. In this heatmap we classify timesteps based on their distance from the mean, measured in standard deviations. We observe atypically high travel times after 10 PM on several days, with a cluster in late September-early October that likely correspond to construction work. Figure 60 also shows periods of atypically low travel times during the morning and evening peaks, some of which are clusterd around the Thanksgiving and Christmas holidays, while some happen during the summer. Overall, approximately 80 percent of the time intervals corrspond to typical conditions as defined above. Figure 61 presents the distribution of atypical values.

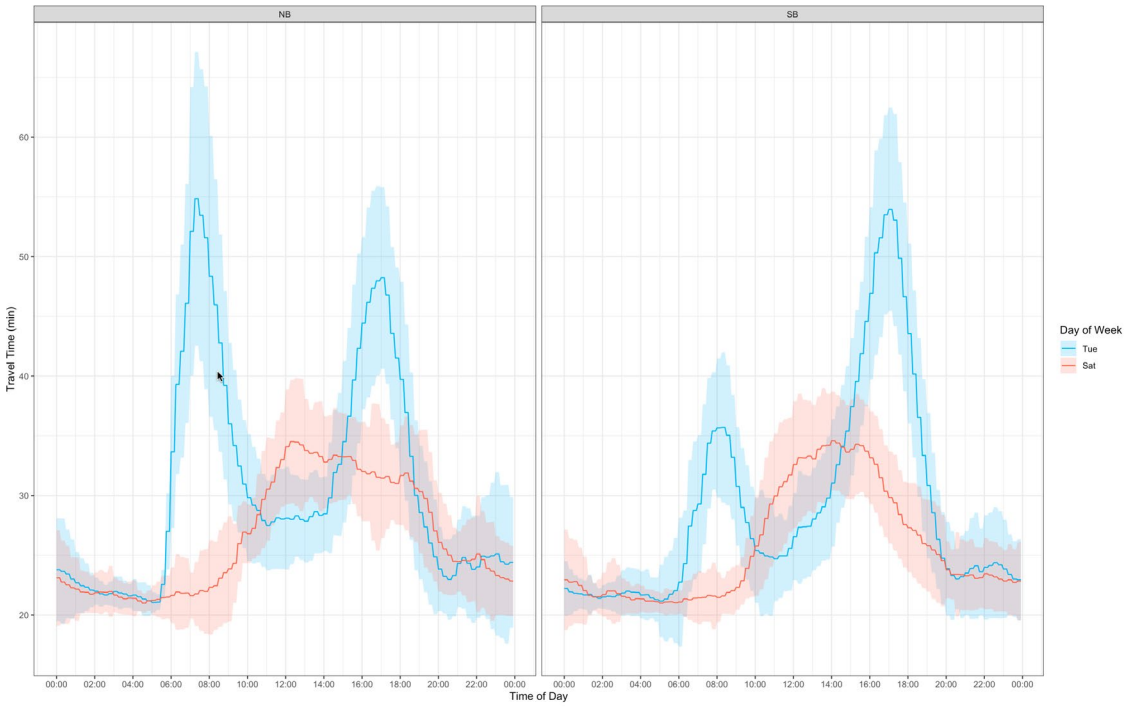


Figure 59. Average 15-minute travel-time and corresponding standard deviation for 2 days of the week (2019).

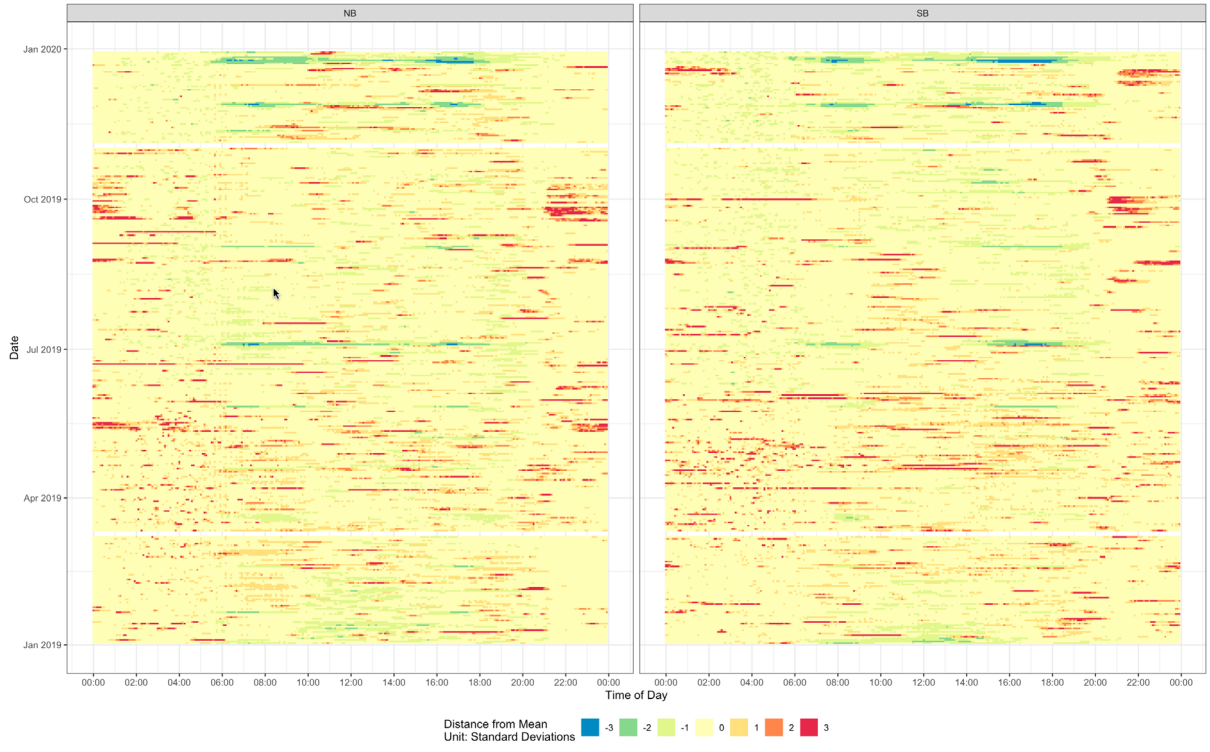


Figure 60. Difference between average five-minute travel times in 2019 and corresponding mean values for 2019. Negative values reflect day/time combinations in which travel times are lower than expected; positive values represent times during which travel times exceeded typical values.

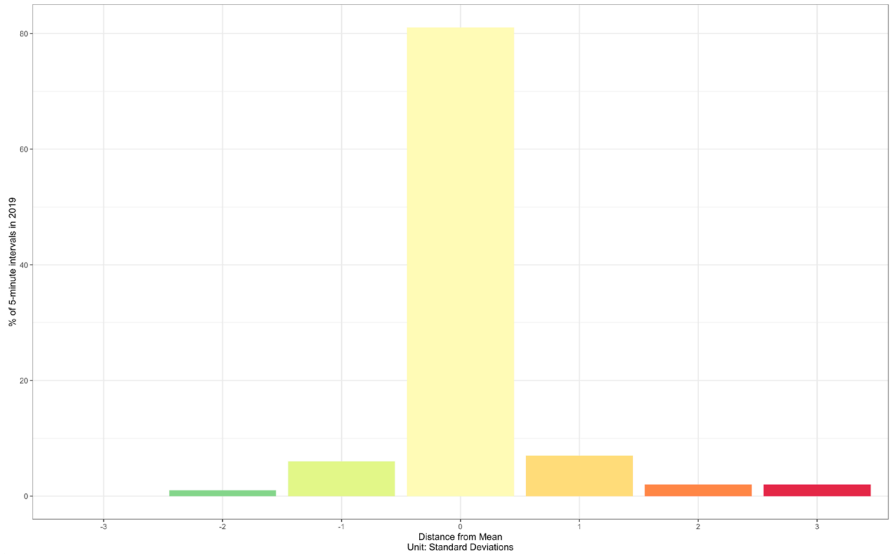


Figure 61. Distribution of 5-minute travel times based on their proximity to the corresponding average (expected) value in 2019. The "0" category corresponds to travel times that are within one standard deviation of the expected value and may be considered "typical" values.

10.1.2.2. Smart work zone trailer data

TxDOT's Austin District placed smart work zone trailers (SWZT) along I-35 to support construction work starting in late 2017. Researchers began archiving the corresponding data in early 2018 for a separate research effort. The archived dataset includes speed and volume information at the location of each sensor, provided every minute. This study considers 2019 data for 17 sensors in the northbound direction and 16 sensors in the southbound direction (Figure 56). Only volume data was considered. There were some missing data points in the dataset due to sensor/communication malfunctions. Intervals with missing data were identified and discarded in the training process.

10.1.2.3. Data preprocessing

The 85 INRIX sensor readings, 33 SWZT sensor readings, and 33 SWZT sensor data availability indicators were tabulated for the 105,120 five-minute intervals in 2019. For the purposes of model development, a day was assumed to start at 3 AM and end at 3 AM the next day. Because traffic tends to be the lightest and most people tend to be at home, 3 AM is generally considered the beginning of the daily traffic cycle in most travel demand modeling studies at this time. This is relevant in the context of travel time prediction because the predictions made closer to 3 AM would be least dependent on the earlier travel times, an assumption that formed the basis for structuring the inputs of the Recurrent Neural Network (RNN) model. Records from the days of March 9, March 10, November 2, and November 3 were removed from the dataset because the time changes for daylight saving time occur on March 10 and November 3 and there was not a

consistent way of using data from these dates with the RNN model. Data from December 31 was also removed because data until 3 AM the following day was not available.

10.1.2.4. Train-test split

Since models that have a very high number of parameters tend to overfit to the data used for training, the available data was split into two subsets in a 4:1 ratio. The larger of the subsets was used for training and validating (if required) the models. The smaller subset was set aside for evaluating the model accuracy. The dataset was split at the day level and not at the five-minute timestep level. Splitting at the day level was deemed necessary because the travel-time patterns do not vary much between nearby five-minute timesteps. Therefore, randomly splitting the data at the timestep level would cause the records in the testing dataset to be very similar to the records in the training dataset, which was not desirable. Therefore, the data from the 360 (365–5 days removed) days in the dataset was split randomly into a training and validation dataset that has all the records from 288 of the days and a testing dataset that has all the records from the other 72 days.

10.1.3. Model development and training

A time series (TS) model and a recurrent neural network (RNN) model were trained to forecast the segment-level travel times up to twelve timesteps (one hour) into the future using only the data available until the current timestep. Separate models were estimated for forecasting the travel times for each of the 85 road segments for each of the twelve future timesteps. That is, for each model type (TS and RNN) a total of models were estimated. The segment-level travel times were assumed not to vary after the twelfth timestep. The forecasts were then used to compute the dynamic corridor-level travel times.

10.1.3.1. Auto-regressive timeseries

The autoregressive timeseries model expresses the target variable—the travel time of a segment at a future timestep—as a linear function of all the predictors. Predictors include INRIX segment travel times and SWZT volume data and corresponding data availability indicators from the most recent six timesteps (30 minutes). For each travel direction we consider data from all sensors on the same direction as the target segment. Additionally, dummy variables denoting the day of the week were also used as predictors. They included binary indicators for each five-minute timestep of Friday, Saturday, and Sunday. The dummy variables encoded for the other four weekdays did not distinguish between days of the week. That is, each five-minute timestep in either Monday, Tuesday, Wednesday, or Thursday had the same dummy variable. The dummy variables for the first four weekdays were collapsed in this manner because the travel-time patterns for these days were relatively similar. The time series model

for each northbound segment had a total of 6 timesteps x (43 INRIX sensors + 16 SWZT sensors + 16 SWZT sensor availability) + 4 days x 24 x 60/5 = 1608 predictors. The timeseries model for each southbound segment had a total of 6 timesteps x (43 INRIX sensors + 16 SWZT sensors + 16 SWZT sensor availability) + 4 days x 24 x 60/5 = 1602 predictors. The coefficients of the predictors were estimated using the scikit-learn package in Python (Pedregosa et al., 2011).

10.1.3.2. Recurrent neural network model

The recurrent neural network model belongs to a class of neural networks called sequence networks (Sherstinsky, 2020). As the name suggests, these networks are used to make predictions when an ordered sequence of data is provided as an input. Each set of inputs in the sequence is passed through the same network. The activations of the hidden nodes or hidden states in the network after the passage of one set of inputs is used as input for the next forward pass through the network. The basic RNN architecture can be schematically represented as shown in Figure 62. In this figure, $x_1, x_2, \ldots, x_t$ are the inputs in sequence. For the travel time forecasting RNN, each x_t would be the sensor readings, sensor availability indicators and the time-of-week variable at timestep t. y_t is the target variable at timestep t which in this case would be the travel time of a segment f timesteps into the future (segment-level travel time at timestep $t+f$). The hidden states h_t store information from the past timesteps that would be relevant for making predictions in future timesteps.

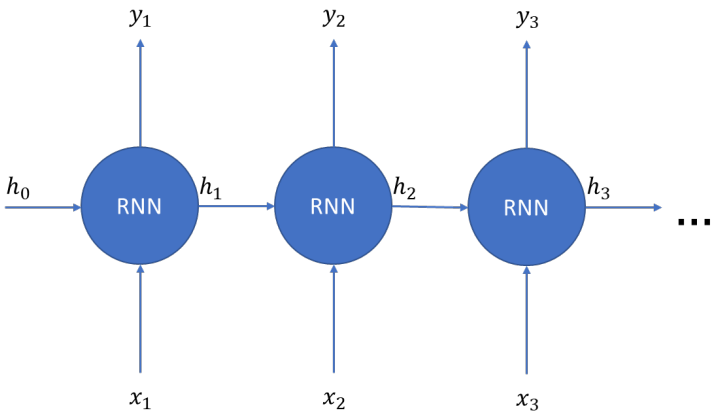


Figure 62. General structure of recurrent neural networks.

While the sequence of inputs provided to an RNN can be arbitrarily long, training RNNs with lengthy sequences can be computationally intractable. Therefore, we trained the RNN using data over a 27-hour period. Each sequence consisted of data over the 24-hour period of a day (which starts at 3 AM) and a three-hour period before the start of the day. That is, for each day, the inputs are supplied from midnight until 3 AM the next day. The hidden states of the network at the timestep corresponding to midnight are all set to zero. The outputs produced by the RNN for

the first three hours are not used because the hidden states at these timesteps may not have captured enough past information to make accurate predictions. At 3 AM (the same day), the hidden states are considered to be set and the predictions from the RNN are used.

10.1.3.3. Training and hyperparameter tuning

A large number of hyperparameters are available that control the RNN architecture and the training process. To determine reasonable network architectures and training hyperparameters, the data that was not used for testing was further split into a smaller training dataset and a validation dataset. The training dataset included data from 230 randomly selected days and the validation dataset included the data from 58 randomly selected days (4:1 split). Several combinations of network architectures and training parameters were tested. Each hyperparameter combination was used to train a subset of the networks (corresponding to a few of the future timesteps) on the training dataset and their prediction accuracy was evaluated against the validation dataset. The set of hyperparameters that produced good results across the subset of networks was used to train all the networks corresponding to the different segments and future timesteps.

The Adam optimization algorithm (Kingma & Ba, 2017) was used to estimate the weights of the network. In each iteration of the Adam optimizer, one day's (27 hours of input and 24 hours of output) data was passed through the network and the network weights were adjusted. An initial learning rate of 0.001 was used for training. The beta hyperparameters of the Adam optimizer were not changed from the default values of 0.9 and 0.999. The model is considered to have been trained for one epoch when all the data in the training dataset has been iterated once. After each epoch of training, the model accuracy was tested against the validation dataset. The training of each network was continued until the network failed to show any improvement in accuracy with respect to the validation dataset over several epochs.

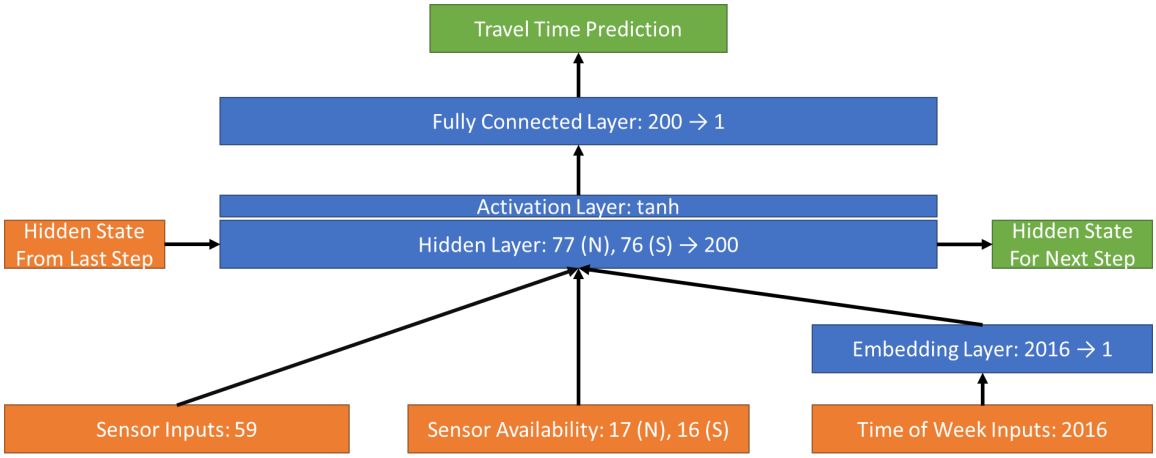


Figure 63. RNN architecture.

The finalized RNN network architecture is shown in Figure 63. The time-of-week is indicated by a single integer variable with 2016 () possible values each of which represents a 5-minute period in a week. The day-of-week variable is passed through an embedding layer, which produces a single scalar output. The output produced by passing a categorical variable through this embedding layer would be equivalent to the output produced by a linear function on the dummy variables of the categorical variable. The sensor inputs (42+17 for northbound segments and 43+16 for southbound segments) and the sensor availability data are also passed as input. Overall, the input to the RNN at a single timestep has 76 (northbound) or 75 (southbound) numerical predictors and 1 categorical predictor with 2016 categories. These inputs are in addition to the hidden states that are passed over from the previous timestep.

The inputs after embedding the time-of-week pass on to a hidden layer which consists of 200 hidden nodes. The tanh activation function is used for this layer. The activations of these nodes are passed on to the output layer which performs a linear operation on the activations and produces a single output. The activations are also passed as hidden states to the RNN for the next timestep.

Recently, more advanced sequential network architectures, such as the long short term memory (LSTM) and gated recurrent units (GRU), have gained traction in the deep learning literature (Cho et al., 2014; Sherstinsky, 2020). We evaluated the performance of an LSTM architecture in the initial tests. Although the LSTM network produced slightly more accurate results with respect to the validation dataset, the increased time required for training LSTM networks led us to prefer the RNN architecture for the final model.

The RNN model was developed and trained using the PyTorch library in Python (Paszke et al., 2019). An Nvidia GeForce RTX 3090 graphics processing unit (GPU) was used to accelerate the training process. The initial training was completed in approximately eight hours. To fine tune the model weights further, training was continued at a reduced initial learning rate of 0.0001. This fine-tuning step took another nine hours. In hindsight, this fine-tuning step may not have been necessary since much of the model improvement occurred in the first eight hours.

10.1.4. Model evaluation

The prediction accuracy of the models developed in the previous section was evaluated on the test dataset. The models were compared in terms of the accuracy of their short-term segment-level forecasts as well as their corridor-level dynamic travel-time predictions.

10.1.4.1. Short-term segment-level travel time forecasts

The static RNN and TS models were used to produce segment-level travel-time forecasts for the 85 time segments for up to twelve timesteps into the future. The forecast produced by the

static model for a segment was simply the last recorded travel time of the segment. The mean squared errors (MSE) of the predictions are plotted in Figure 64. The MSEs of the segments with the same direction were averaged. The MSE was computed as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y)^2,$$

where n is the number of timesteps in the test dataset, $\hat{y}_i$ is i^{th} travel time forecast, y_i is i^{th} actual travel time.

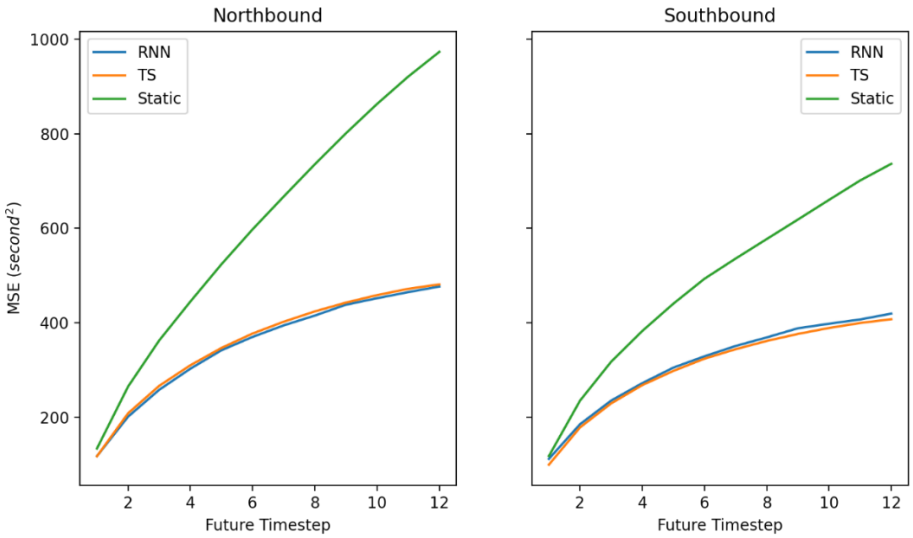


Figure 64. Short-term segment-level forecast MSE comparison

The RNN and TS models have much lower MSEs than the static model. While the RNN model produces marginally better results than the TS model for the northbound segments, the opposite is true for the southbound segments. As expected, all the models produce worse predictions for timesteps that are further into the future.

10.1.4.2. Corridor-level dynamic travel times

Using the short-term segment-level travel-time forecasts, the corridor-level dynamic travel times were computed. The dynamic travel times were compared against the actual travel times computed using the known travel times of each segment at each timestep. The MSE and the mean absolute error (MAE) of the dynamic travel times computed using forecasts from the different models are shown in Figure 65. The MAE was computed as

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y|,$$

where n is number of data points, $\hat{y}_i$ is i^{th} travel time prediction, y_i is i^{th} travel time.

For a sense of scale, the average travel time along a direction of the corridor was around 1700 seconds. Once again, the RNN model performs slightly better than the TS model in the northbound segments while the TS model performs slightly better in the southbound segments. The MSEs of the TS and RNN models are more than 40 percent lower than that of the static model. The MAE is more than 20 percent lower.

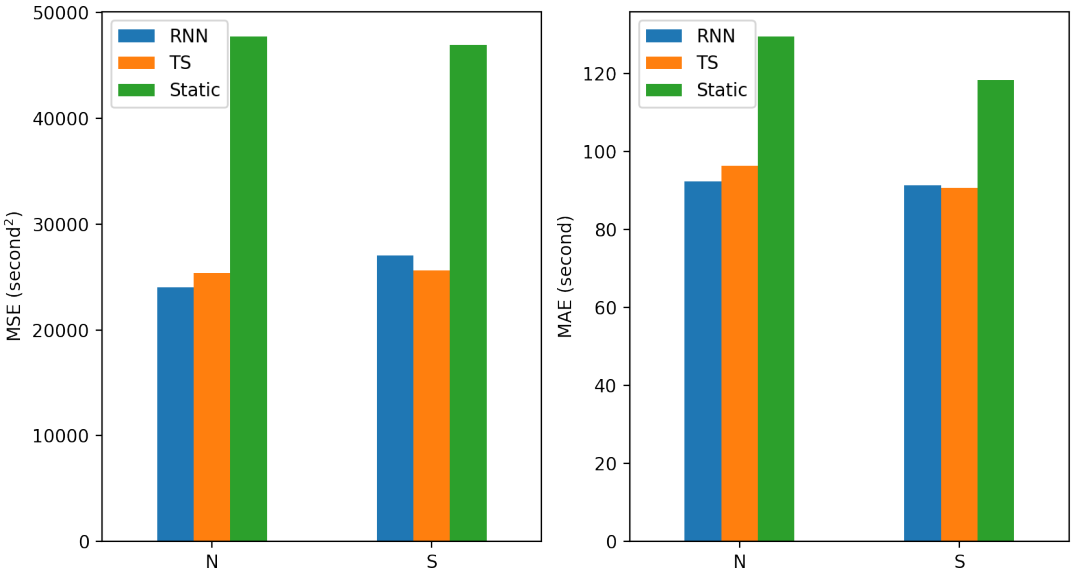


Figure 65. Dynamic travel-time mean squared error and mean absolute error.

The differences between the prediction accuracies are even starker when considering only the timesteps in the peak period between 6:30 AM and 9:30 AM and the period between 3:00 PM and 7:00 PM on weekdays (see Figure 66). In this case, the reduction in MSE from the static model is more than 50 percent and the reduction in MAE is more than 35 percent.

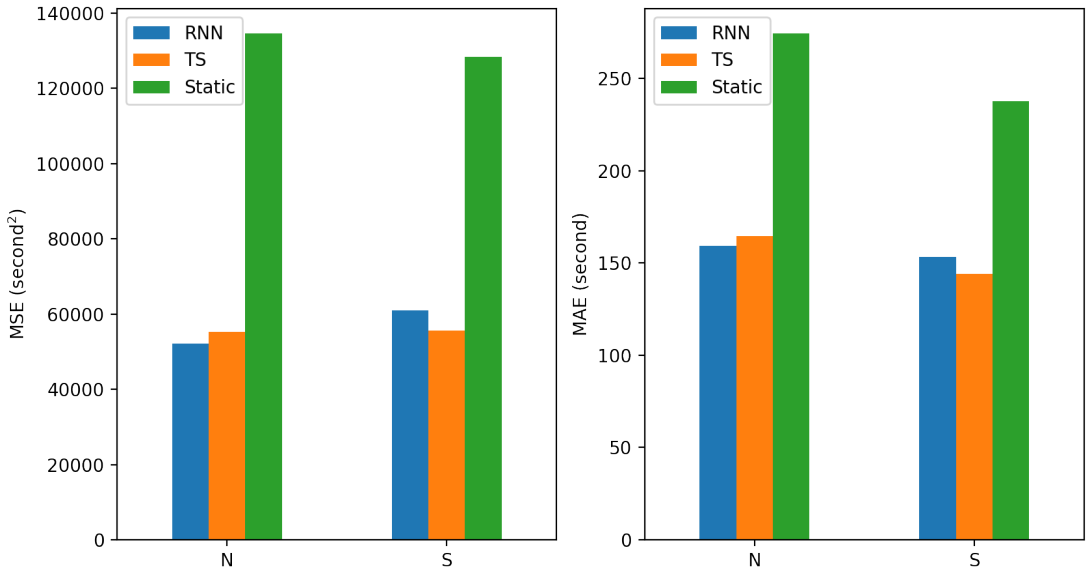


Figure 66. Dynamic travel-time mean squared error and mean absolute error during peak periods.

10.1.4.3. Analysis of model forecasts over one year of data

This section discusses the accuracy of model forecasts when all the days in 2019 for which data is available are considered. Some of these days were part of the training dataset, but the purpose of this section is not to evaluate model performance but rather to identify trends and patterns in the observed performance that may support future model improvement and refinements.

Figure 67 presents the prediction error for all considered five-minute intervals in 2019 in the northbound direction (similar patterns are observed in the southbound direction). The figure illustrates that both analyzed machine learning models perform better than the static approach, which has a strong tendency to underestimate travel times as congestion builds and to overestimate travel times when congestion dissipates. Figure 67 also show clusters of intervals during which prediction errors are very high across methods. Many of these coincide with the atypical traffic conditions illustrated in Figure 60.

Figure 67. Prediction error by five-minute interval for all analyzed methods (northbound).

Figure 68 further illustrates how the static model fails to anticipate peak-period traffic patterns by presenting the northbound travel times on January 29, 2019. There are two peak periods centered around 8 AM and 5 PM respectively. The static model underestimates the travel times at the beginning of these peak periods because the model does not anticipate the increase in traffic. Similarly, the static model overestimates travel times toward the end of the peak periods because the model does not anticipate dissipation of traffic. The RNN and TS models can account for the time-based traffic characteristics because time-of-week variables were used as predictors.

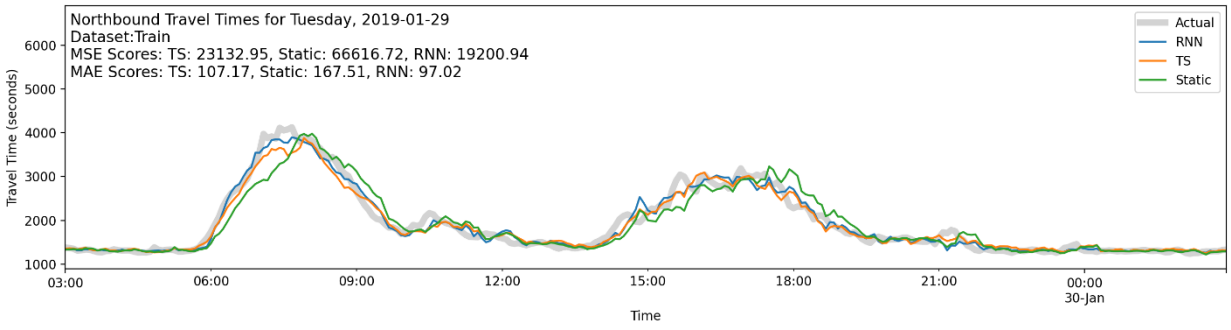


Figure 68. Travel times on January 29, 2019.

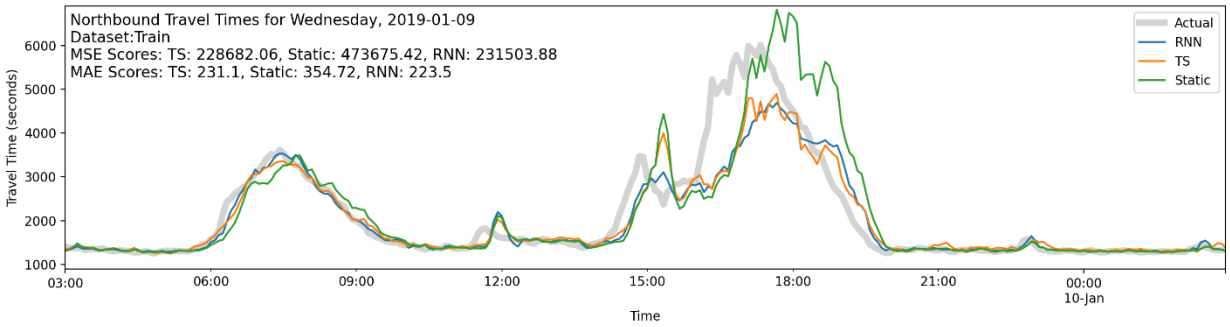


Figure 69. Travel times on January 9, 2019.

In general, the static model fails to anticipate the dissipation of traffic after a period of unusually high travel times. For example, consider the predictions made for January 9, 2019, for segments in the northbound direction (see Figure 69). The high travel times that occur around 5 PM on this day is unusual. Once these high travel times are recorded, the static model expects the high travel volumes to persist into the future while the RNN and TS models can recognize that the travel times are likely to reduce especially since the peak period would also be coming to an end. However, note that none of the models are able to anticipate the initial occurrence of the unusually high travel times at around 4 PM.

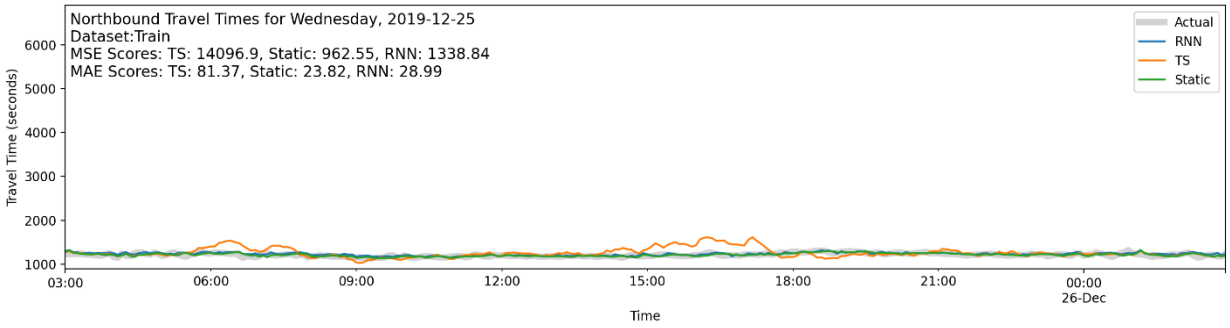


Figure 70. Travel times on December 25, 2019.

Figure 71 and Figure 72 present the distribution of prediction errors during typical and atypical conditions, respectively. These figures correspond to the northbound direction, but similar patterns are observed in the southbound direction. During typical traffic conditions, which represent 80 percent of the observed conditions in 2019, prediction errors are centered around zero for all models. The static model has heavier tails, which indicate a larger number of instances on which prediction errors are far from zero.

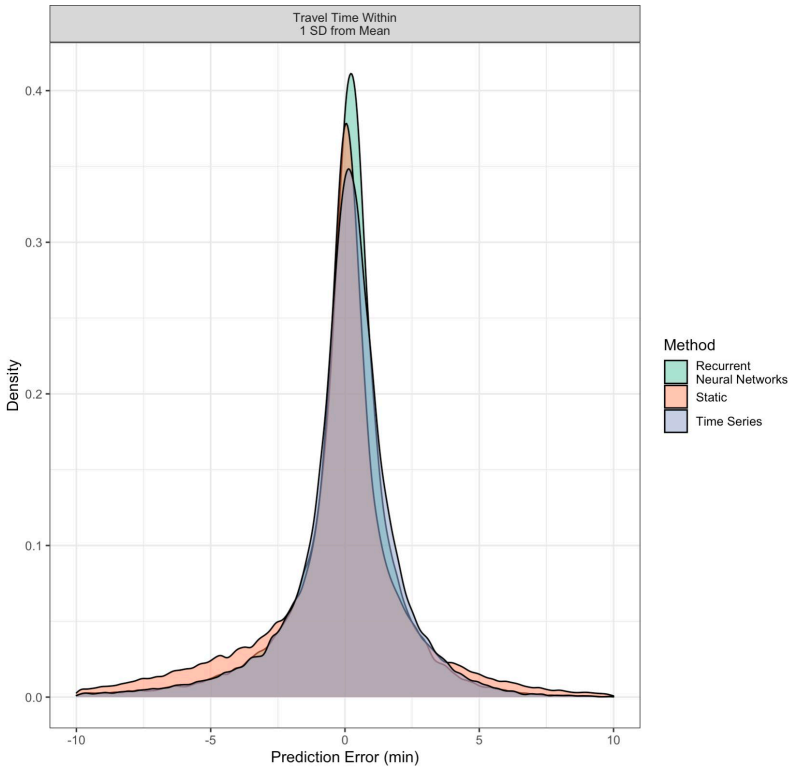


Figure 71. Distribution of prediction errors during "typical" traffic conditions.

The patterns for atypical conditions are different depending on whether travel times are higher or lower than typical. When travel times are higher, the distribution becomes flatter; it remains centered on zero when the deviation from average remains below two standard deviations. Forecasts for time intervals during which the travel time exceeded typical values by more than two standard deviations exhibit higher errors, with all models being more likely to overestimate travel times under these conditions.

Figure 73 illustrates an instance in which the RNN and TS models performed poorly. On January 15, 2019, around 6:45 AM, a fatality caused by the collision of an 18-wheeler and a pedestrian caused all lanes along portions of the road to be temporarily closed (Kamath, 2019). During this incident, the static model produced much better predictions than the timeseries model, which tended to overestimate the travel time, and the RNN model that underestimated travel times.

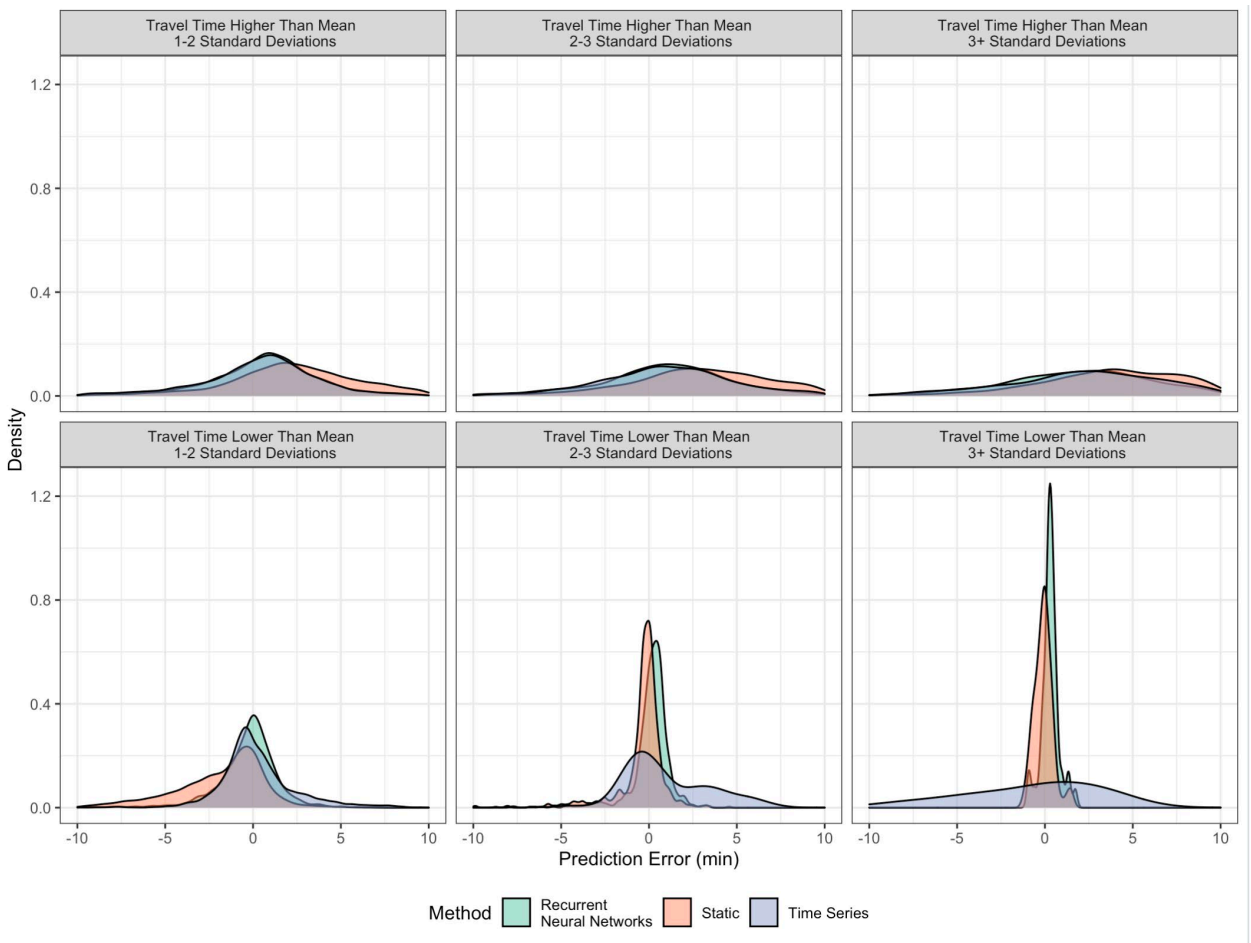


Figure 72. Distribution of prediction errors during "atypical" traffic conditions.

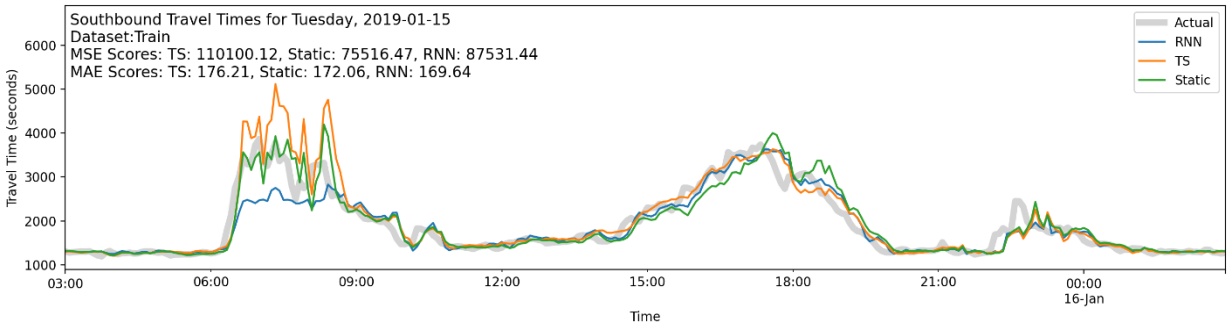


Figure 73. Travel times on January 15, 2019.

Static and RNN models perform well when travel times are atypically low, but TS models tend to overestimate travel times. TS models are also observed to forecast travel time increases because of peak traffic even on national holidays (note that the models do not use any predictors that explicitly indicate holidays). Consider the predictions made for Christmas day in 2019 (see Figure 70). The TS model continues to predict a slight increase in travel time around 7 AM and 4 PM corresponding to the peak travel times that typically occur around those times on

a Wednesday. The static model performs well in this case because there is hardly any variation in travel times throughout the day. The RNN does not predict peak traffic despite the day being a Wednesday probably because the RNN model can handle non-linearities and has short-term memory. Therefore, the model probably realizes that travel times will not increase suddenly in the peak period if travel times have been lower relatively throughout the rest of the day.

Figure 74 summarize the root mean squared error during typical and atypical traffic conditions by time of day. The atypical conditions visualized in this figure correspond to cases in which actual travel times are between one and two standard deviations from the corresponding average value in 2019. Results suggest that during typical conditions RNN and TS models perform very similarly and lead to lower errors than the static approach, particularly during the peak periods. When traffic conditions are milder than usual (i.e., actual travel times are lower than average) RNN models perform very well, but TS models and the static approach tend to overestimate experienced travel times. Prediction errors are larger for all models when actual travel times are higher than average, but RNN and TS models perform better than a static approach.

Figure 75 analyzes the distribution of actual error values by time of day in the northbound direction (similar patterns are observed in the southbound direction). This figure reflects the patterns observed in Figure 67, suggesting that the static method tends to underestimate travel times during congestion build-up and to over-estimate them during congestion dissipation. The errors are larger during atypical conditions, particularly when travel times are higher than usual. The error patterns are similar for RNN, but error values are significantly lower. TS models tend to overestimate travel-time predictions during atypically low congestion.

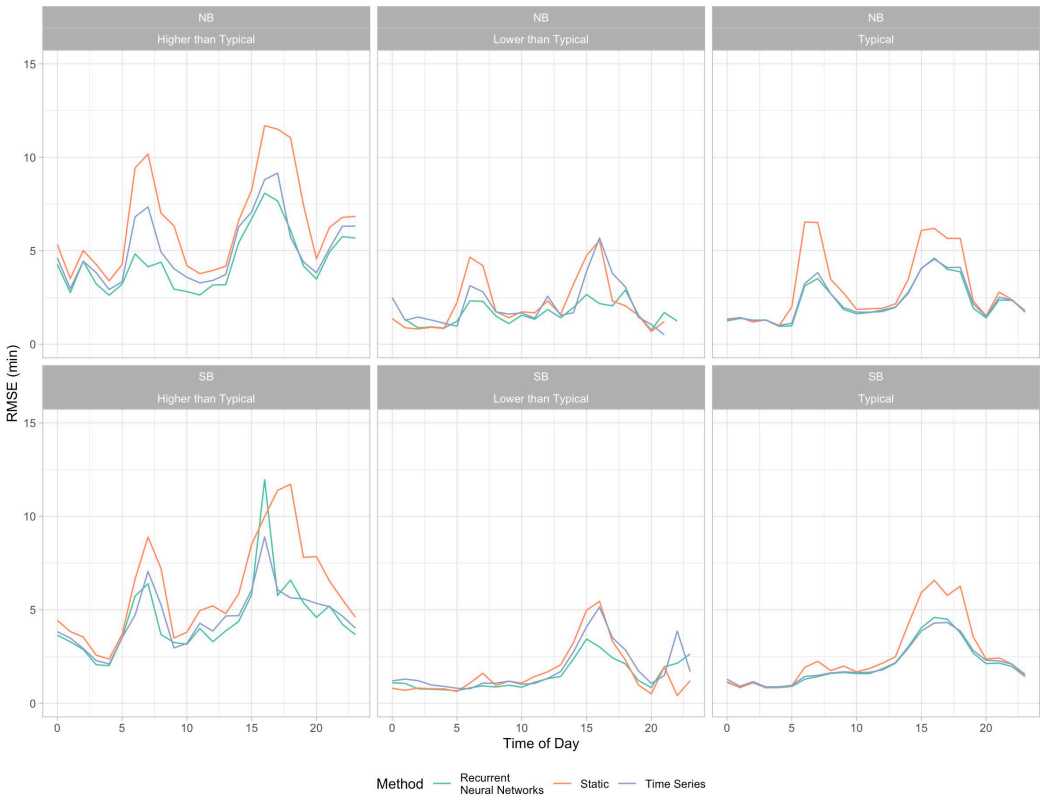


Figure 74. Average prediction error by time of day under different typical (actual travel times within one standard deviation from the mean) and atypical (actual travel times between one and two standard deviations from the mean) traffic conditions on weekdays.

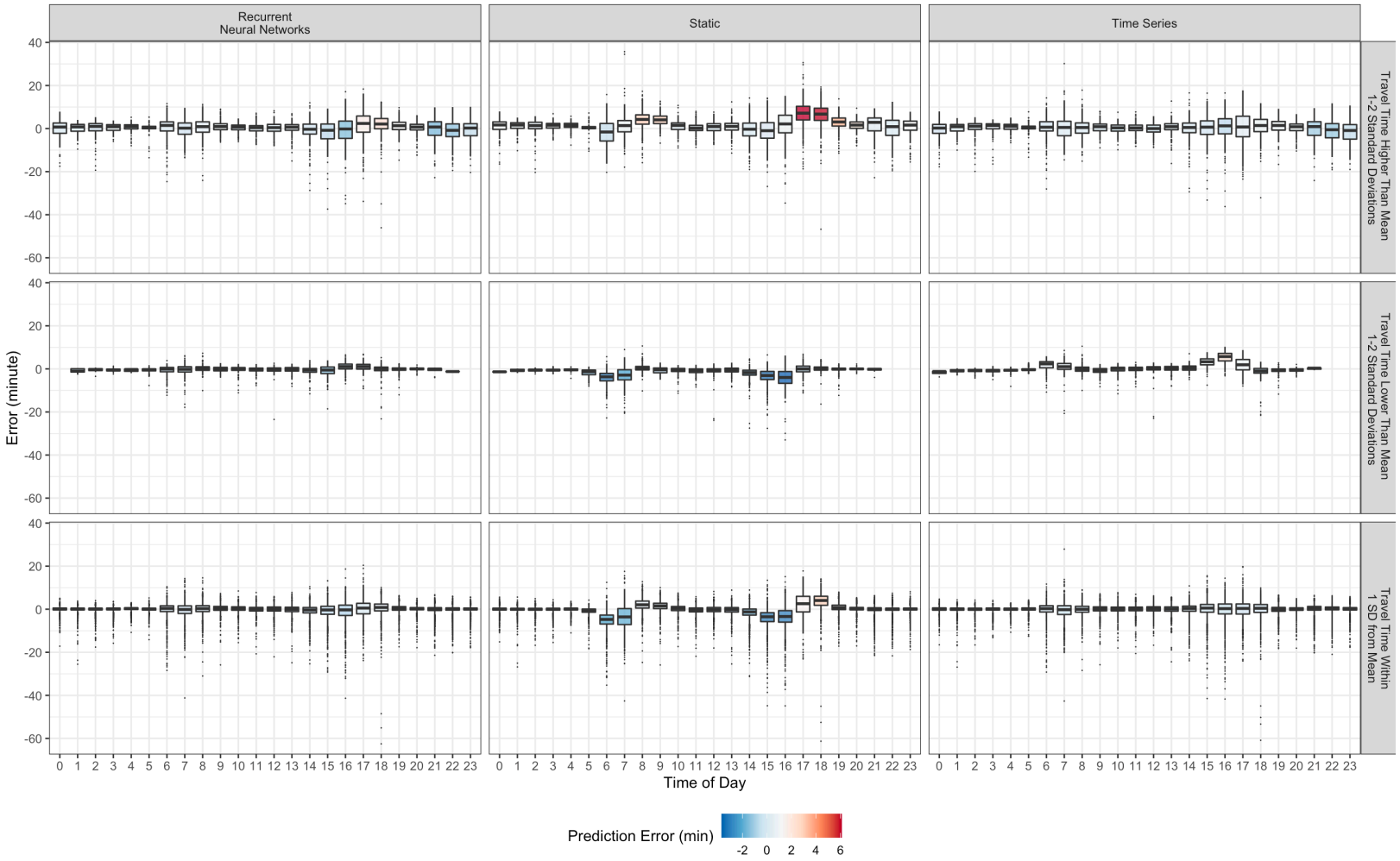


Figure 75. Prediction error by time of day for typical and atypical traffic conditions (northbound direction, weekdays).

10.1.5. Discussion

A methodology for computing corridor-level dynamic travel times using short-term forecasts of segment-level travel times was developed. The short-term travel-time forecasts were made using recurrent neural networks (RNN) and time series (TS) models. The accuracy of the forecasts and the resulting dynamic travel times were compared against that produced by the static model, which does not consider the temporal variability of segment-level travel times. Overall, the travel times predicted by the TS and RNN model were more accurate than the static travel times. The mean squared error of the TS and RNN models were more than 40 percent lower than that of the static model. Also, the mean absolute errors of the TS and RNN models were more than 20 percent lower. There was no clear winner between the TS and RNN models. While the RNN model performed slightly better on the northbound segments, the TS model was better on the southbound segments. On public holidays, the RNN model performed better than TS. But training the RNN model required more of the modeler's time to identify a suitable network architecture and acceptable hyperparameters.

This document also investigates model performance by time of day, and on days with atypical traffic conditions. Our findings suggest that the benefits of TS and RNN models with respect to a static approach are more obvious during the peak periods and when traffic conditions are atypically high. When conditions are atypically low, RNN models provide accurate travel-time estimates, while TS models tend to overestimate expected travel times, particularly during the peak period.

The large number of hyperparameter values, network architectures, and high training times make a comprehensive evaluation of neural network approaches impossible in a limited timeframe. A few promising avenues for further improving the accuracy of the neural network model would involve: (1) using different variants of RNN, such as long-short-term-memory (LSTM) and gated-recurrent-unit (GRU), and (2) training with more than one year of data. More advanced timeseries models such as the seasonal auto-regressive integrated moving average (SARIMA) model may also be tested. Additionally, including indicators that reflect the deviation of observed travel times with respect to average conditions, or the presence of traffic incident or other type of roadway closures, may improve model performance under atypical conditions, which account for approximately 20 percent of the observed conditions in 2019. Data from more years may be necessary so that a good number of days with special conditions are present in the training dataset. Another important consideration to evaluate is the potential benefits of the proposed models for TxDOT planners and operators is the frequency at which forecast will be updated, which is highly dependent on the specific use case.

10.2. Development of Models for Field Testing

This section describes the development, testing and prototype real-time deployment of machine-learning (ML) models for short term travel time prediction on eight bi-directional corridors at two test sites. One of the goals of the work presented in this chapter is to understand how some of the models evaluated in Section 10.1 perform at different locations with varying data availability, and to gain further insights into the value of short-term travel time prediction models. Another important objective of the work presented here is to prototype a feasible implementation framework that can be replicated at multiple sites. Researchers developed partially automated workflows to streamline model development and testing and built a prototype data pipeline to deploy the models and share real-time results with TxDOT using a web-based application. The software tools, data pipelines and web-based application are described in detail and documented in Products 5, 6, 8 and 9.

Section 10.1.10 presents the experimental design and Section 10.1.3 describes model development and testing, which includes two phases that use slightly different assumptions for models specification. Section 10.3 describes the evaluation framework and provides a summary of the overall findings. We provide a qualitative and quantitative assessment of the value of short-term travel time prediction models in Section 10.3.7 and propose a pathway to implementation in Section 10.5.

10.2.1. Experimental Design

This section introduces the sites selected by CTR for the deployment and testing of short-term travel-time prediction models and provides an overview of the experimental design and workflow.

In the context of this project, model deployment involves training previously developed machine learning (ML) models at new sites and developing data pipelines to produce real-time travel time estimates using such models. Travel times will be computed using CTR resources and delivered through a web interface that may be accessed by TxDOT.

Section 10.2.1.1 describes the selected sites; Section 10.2.1.2 provides an overview of the required data workflows and framework components.

10.2.1.1. Site selection and data sources

CTR identified two sites for model testing in collaboration with the TxDOT project team, one in Austin and one in El Paso. At each site, models are trained and tested for one main corridor. Once the workflows for model training and testing were streamlined, CTR performed a similar analysis on a second corridor. At both sites, secondary corridors act as an alternative route to

the primary corridor. TxDOT is currently providing (or planning to provide) comparative travel-time information between each pair of corridors at critical decision points to support travelers' route selection process. In this context, travel-time estimates that are close to experienced travel times may be particularly important to encourage drivers to trust and act on the provided information. Figure 76 and 77 present the primary and secondary corridors at each site. In El Paso, CTR analyzed I-10 between Horizon Boulevard and Antonio Street as the primary corridor. The secondary corridor on this site is FM 375 between Gateway Boulevard and Desert Boulevard. In Austin, CTR analyzed SH 130 between its intersection with I-35 near FM 1375 in Buda and its intersection with I-35 north of Georgetown. I-35 between Turnersville Road in Buda and SH-195 near Georgetown acts as a secondary corridor at this site.

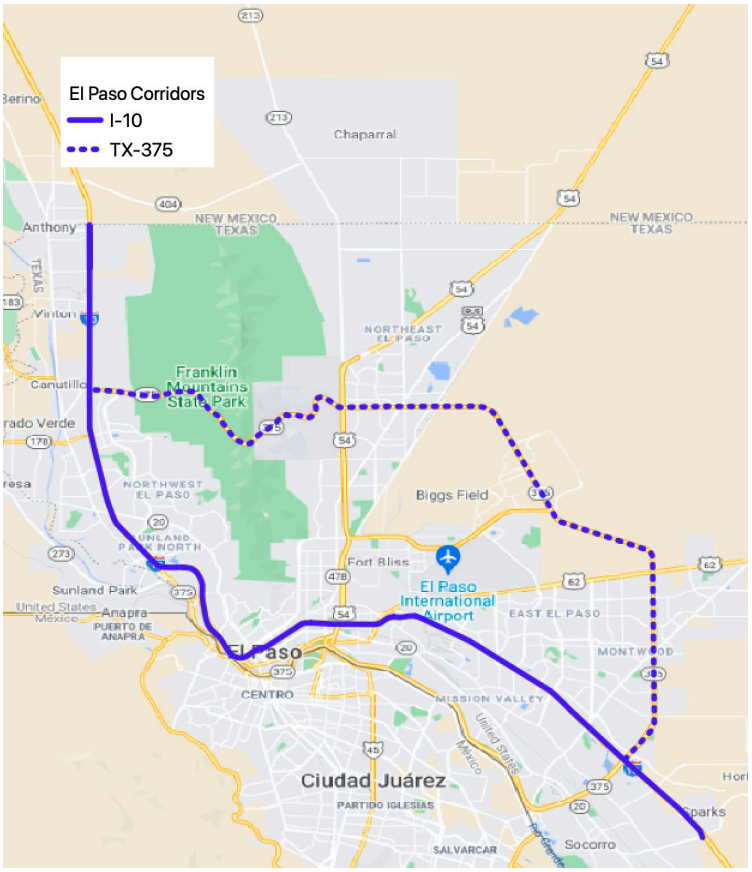


Figure 76. Primary and secondary corridors in El Paso.

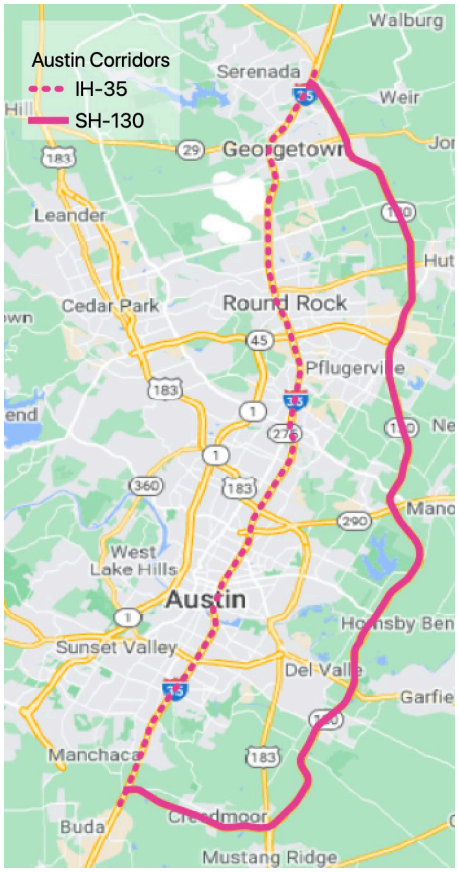


Figure 77. Primary and secondary corridors in Austin.

Figures 78 and 79 illustrate the current position of dynamic message signs (DMS) on the two sites considered for implementation.

The main data source to be used at both test sites consists of probe-based speeds, currently available to TxDOT through INRIX. It is also desirable to include traffic volume data in the model specification, although such data is not as widely available as speed information and is also relatively less standardized. This project utilized data from permanent and semi-permanent (e.g., smart work zone trailers) ITS devices that stream data through Lonestar. Challenges with the use of this data will be discussed later in this report.

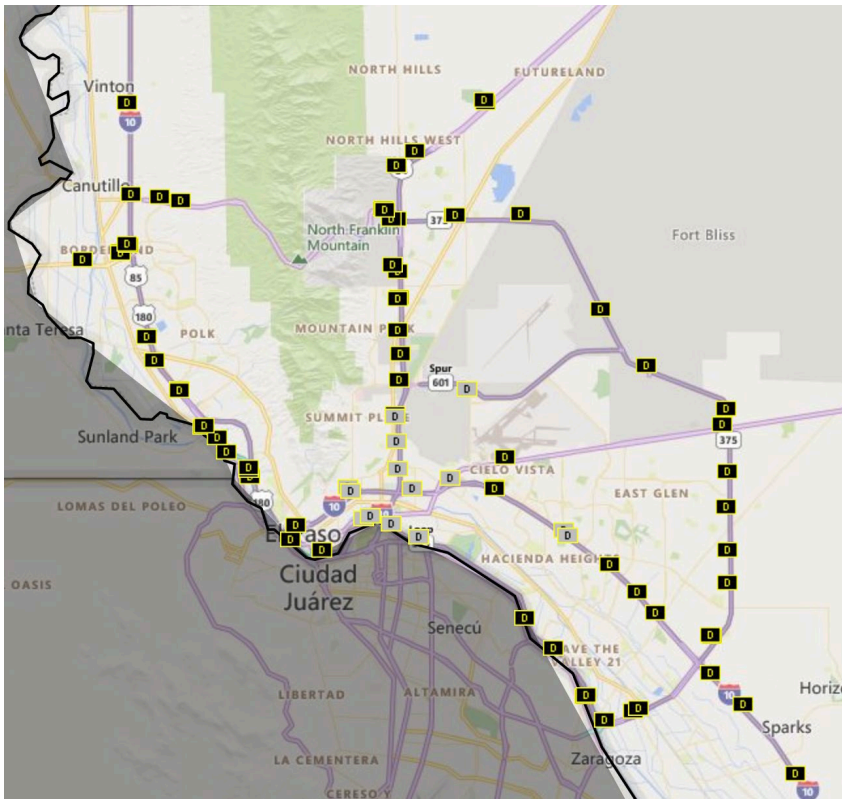


Figure 78. Location of existing DMS on I-10 and FM 375 through El Paso District.

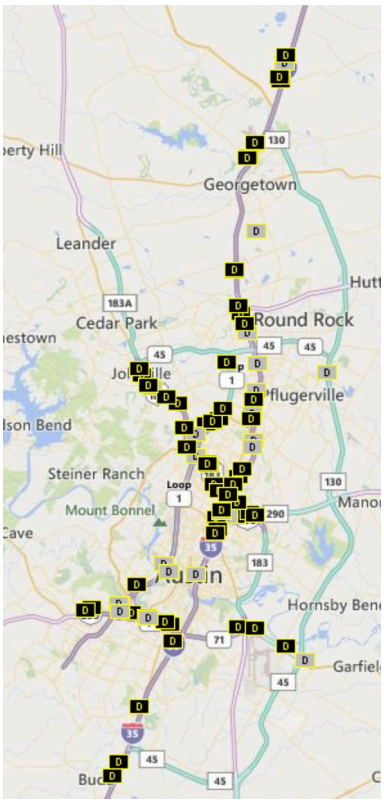


Figure 79. Location of Existing DMS on I-35 and SH 130 in the Austin District.

10.2.1.2. Data Workflows and Framework Components

One of the goals of this chapter is to develop workflows to support systematic model development and testing, and to create a prototype framework to support real-time deployment.

The following steps are required for model development and testing and were partially automated by CTR:

- Collect/retrieve historical data for model training.
    - **INRIX data:**
        - Identify INRIX segments corresponding to all selected corridors and obtain their metadata. This step is done manually using INRIX web interface.
        - Download data samples for model training. CTR developed an automated process to download data using an API provided by INRIX and downloaded segment metadata. The API was discontinued in August 2023, and this process will have to be conducted manually through the web interface.
    - **Volume data:** CTR developed workflows to archive volume data from Lonestar. This data has been used for model refinement, though there are outstanding issues with data quality.

- Train models
    - Most trained models use one year of historical probe-based speed data and traffic volume data, where available.
    - Models are trained and tested using the same process implemented for the experiments reported in Section 10.1. Models are trained using scripts that are run manually and use data staged on a database. Training results can be stored in the database to support performance evaluation.
        - Where both volume and travel time data are available, multiple ML models were developed—one using only probe-based speed, and one using both data types—and their performance was compared. This approach is used to understand the limitations of models if deployed at locations where traffic volume data is not available.
        - For model testing and training, CTR assumed that experienced travel times are given by the dynamic travel times computed using probe-based speeds (see Appendix B).
    - CTR explored different levels of spatial aggregation for model training, ranging from segment-level models (such as the ones describes in Section 10.1) to models that span the entire corridor.
- Develop a framework to streamline model training, evaluation and real-time deployment, thus facilitating model use at new sites. Framework components developed by CTR include:
    - Database schema to efficiently store metadata defining analyzed corridors, segments and sub-sections, along with real-time data and model predictions.

- Data pipelines to retrieve speed and volume data in real-time using appropriate API calls and automated methods to produce travel-time estimates for selected origin-destination combinations using trained ML models. Workflows to evaluate model performance in real-time and over user-defined performance periods.
    - Model evaluation includes a comparison of predicted and experienced dynamic travel times.
- Prototype web interface to share real-time travel time predictions and performance evaluation.

10.2.2. Model Development and Training

The models developed in this effort to forecast travel times on segments along a corridor use segment-level speed data at five-minute intervals (as provided by INRIX) and traffic volume data at five-minute intervals collected permanent count stations. Models with different level of spatial aggregation have been tested in this study, ranging from INRIX segment-level predictions to route level forecasts.

When segment-level predictions are used, real-time speed and volume data retrieved every five minutes are used to forecast segment-level travel times on downstream segments for up to one hour into the future. Using such forecasts, origin-destination travel time estimates for time step *t* are computed using forecasted segment-level travel times that correspond to the arrival time at each segment along the corridor, which cannot be measured at time *t*.

Route-level level predictions use real-time segment-level data to predict total origin-destination travel time directly. We also define an intermediate level of spatial aggregation for predictions denoted paths, which are non-overlapping sets of segments that span the desired origin-destination route. When models are trained to generate predictions at the path level, route-level travel times are obtained using the same approach proposed for segment-level predictions.

This study uses recurrent neural networks (RNN), which can capture temporal dynamics. Using the standard Torch Python library, our model creates a RNN for each unit of prediction (segment, path or route) in a corridor for twelve (or more) timesteps into the future, each containing a single hidden layer with 200 nodes. Each of these separately trained models takes as input the per-segment INRIX travel time estimates for all segments in the same travel direction, a vector of volume data along the corridor, indicator variables for the availability/reliability of said sensor data, plus an embedded categorization of the hour of the week from which the data was drawn. These models are then trained on a subset of the available data to minimize the mean squared error (MSE) of the output predictions when

compared to the ground truth, assumed to be the dynamic travel time as determined from INRIX historical data.

The general procedure of developing predictions for the RNN and time-series models is illustrated in Figure 80. The live stream of INRIX travel-time data and Lonestar C2C sensor data is stored in a PostgreSQL database for reference, and the catalogued data is used as training and evaluation data for model development. Additionally, after training has progressed to develop sufficiently performing models, the most recent input data is fed to these models to generate predictions in real time.

For our experiments, we aim to train and test each corridor's models on a year-long collection of data to provide the models with a firm sense of typical traffic dynamics, as well as individual instances of traffic anomalies. Segment-level speed data will be provided by INRIX; CTR has already confirmed access to historical and real-time data, identified the segment IDs corresponding to each corridor, and downloaded data for model training. Traffic volume data is highly desirable, but the availability of live data streams (and corresponding archived data for model training) on the sites selected for this project is limited.

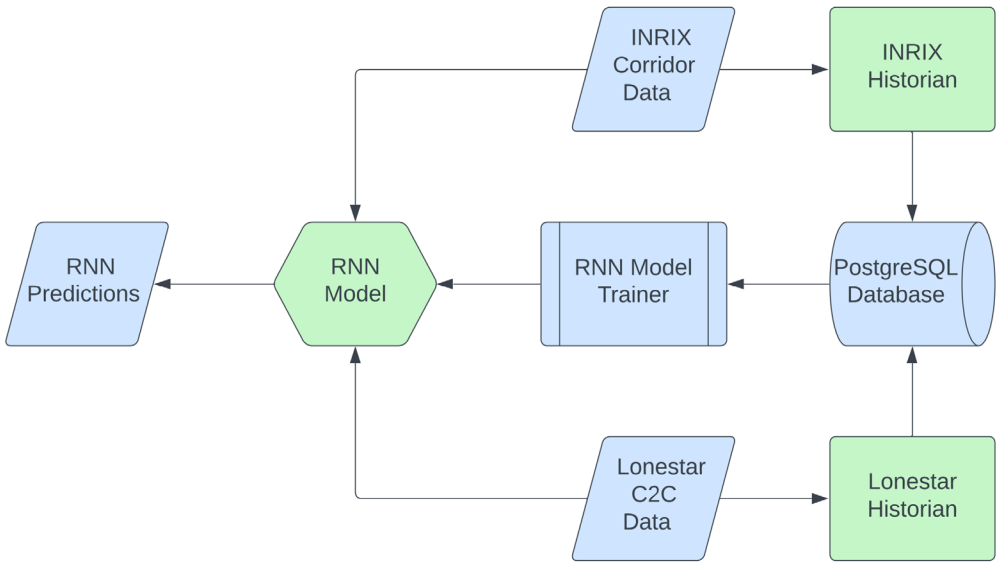


Figure 80: Model training and execution workflows.

Trained models are evaluated by using live-stream data from the INRIX and C2C feeds as inputs to generate real-time origin-destination travel-time estimates. Model outcomes can be archived in a database for model performance analysis and made available in real-time to TxDOT through a simple interface. Travel-time estimates are evaluated against a posterior ground truth obtained from INRIX data. A comparison of the accuracy of the model predictions will be developed, and performance will be evaluated both for long-term accuracy as well as

the ability to accurately predict the onset and dissipation of traffic anomalies. The evaluation process may include comparing model results to the predictions of both our naïve models and to the current estimates provided on dynamic message signs. The model sensitivity to input data quality and availability will also be reported as part of the evaluation process.

The input data received from INRIX and the Lonestar C2C feed require a degree of preprocessing, which we handle in bulk for the RNN model training datasets. Among the critical steps in this is the determination of the UTC and local time at which each data point was collected. This is essential to prevent errors caused by time-zone variability across a given dataset, for example due to daylight saving time. We further embed the timesteps into five-minute bins which reflect the point of time in the current week ("weekpoints"), thus incorporating the cyclic nature of traffic as well as its dependence on daily patterns. This, however, does not include an indicator for longer-term seasonality; thus, the models may not be able to effectively recognize patterns caused by, say, shorter daylight hours in winter.

10.2.2.1. Computational Considerations

While trained models can produce travel-time estimates almost instantaneously, the training process is resource intensive. Given the high quantity of nodes not only in each individual RNN, but also the total number of nodes among all RNNs, it should come as no surprise that the computational complexity of the RNN training task is substantial. The number of parameters to be estimated for a single corridor pair approaches 60 million parameters, which, if trained on a CPU, would require months of wall-clock time to complete. To improve training speed, we offload training to a GPU capable of performing floating point operations at a much higher rate. This reduces the training time for RNNs to a matter of hours. CTR acquired a computer equipped with a high-end GPU capable of reducing training runtimes from days to hours. We further note that the computational complexity of the training appears to scale approximately linearly with both the amount of training data used as well as the number of INRIX segments which make up the corridor for which the RNNs are being trained.

To continue automating the procedure of training and predicting travel times, we developed modularized source code that can be ran as a continuous process (a daemon or app) or on a schedule to further train the model as new datapoints become available, as well to provide an updated travel time upon demand. These processes will communicate with a database to provide data reliability, and a front end will provide predictions to end-users.

10.2.3. Data

The training runs reported in this section considered two data types: segment-level travel times from INRIX and volume data provided by El Paso District from the C2C data feed. Models were trained using either twelve months of INRIX data or six months of both INRIX and C2C traffic-volume data.

For training RNN models in this section, we study the secondary routes along the two bidirectional corridors of focus—I-10 through El Paso and SH 130 bypassing Austin. The development of reliable models for predicting travel times across these four corridors will provide important information for drivers transiting the two studied areas and may influence their route choice significantly. The SH 130 corridor encompasses 109 discrete segments in each direction, with each segment providing independent travel time data. The I-10 corridor provides the same for 60 and 55 segments for the east- and westbound directions, respectively.

INRIX datasets feature incomplete data on occasion, but this is quite rare—less than 0.25 percent of datapoints are missing across all corridors, as shown in Figure 81. Nonetheless, missing data must be accounted for to provide sufficient data for the RNN training. To do this, we make two data manipulations. First, we exclude dates for which a large period of data is missing for all segments in the corridor. These dates tend to involve major events such as the beginning or ending of daylight saving time, or the February 2021 winter storm and ensuing electricity crisis. Second, for data where only a select number of points are missing, we replace missing values with the mean value for the missing segment-weekpoint combination. That is, if a datapoint for segment 100 is missing at weekpoint 20 for a given day, we provide a replacement value calculated as the mean value for all records of segment 100 at week point 20. We further provide to the RNN a Boolean (true or false) variable indicating that this datapoint is unreliable.

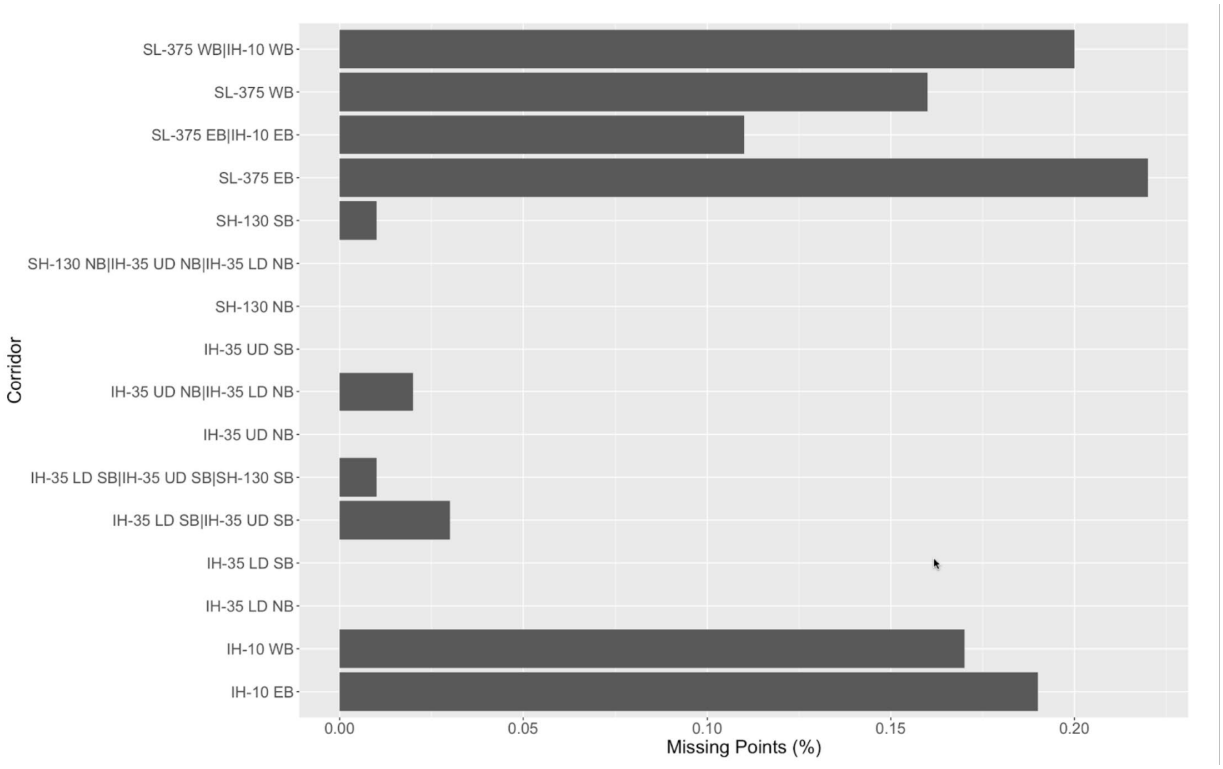


Figure 81: Share of INRIX data missing for each corridor.

Additionally, the C2C data feed also provides volume, occupancy, and speed data at three detector locations, disaggregated across four lanes at each location, for both directions of the I-10 corridor in El Paso. This dataset also includes missing datapoints, with per-lane data availability ranging from 84 percent to 95 percent of datapoints provided depending on detector location and lane. Missing datapoints must also be accounted for, so we preprocess this dataset by replacing missing values with the mean value for that data source across available data for the same weekpoint.

To properly train the RNN models, the utilized dataset must be split into three distinct classes of records: training data, which is fed to the RNN models to develop their parameters in each epoch; validation data, which is fed into the RNN models while holding the parameters constant and provides a scoring metric (in this case, mean squared error) which the RNN seeks to minimize; and testing data, which is only fed into the model upon completion of training as a means to confirm the generalizability of the models. For our models, 20 percent of dates were withheld from the model as the testing dataset; of the remaining data, 80 percent (64 percent of the whole dataset) was provided for training and the other 20 percent (16 percent of the whole dataset) was provided for validation. The dataset was split by date, with each day's data provided from 3 AM to 3 AM the next morning.

The following scenarios were evaluated for route -level predictions (Phase I):

- I-10 Eastbound with 12 months (2021-01–2021-12) of INRIX travel time data
- I-10 Westbound with 12 months (2021-01–2021-12) of INRIX travel time data
- SH 130 Northbound with 12 months (2021-01–2021-12) of INRIX travel time data
- SH 130 Southbound with 12 months (2021-01–2021-12) of INRIX travel time data
- I-10 Eastbound with 6 months (2021-12–2022-05) of INRIX travel time data and C2C volume and occupancy data
- I-10 Westbound with 6 months (2021-12–2022-05) of INRIX travel time data and C2C volume and occupancy data

Based on Phase I results, a second set of models was trained considering different levels of spatial aggregation.

- I-10 Eastbound with 12 months (2022-01–2022-12) of INRIX travel time data
- I-10 Westbound with 12 months (2022-01–2022-12) of INRIX travel time data
- SH 130 Northbound with 12 months (2022-01–2022-12) of INRIX travel time data
- SH 130 Southbound with 12 months (2022-01–2022-12) of INRIX travel time data
- SL 375 Eastbound with 12 months (2022-01–2022-12) of INRIX travel time data
- SL 375 Westbound with 12 months (2022-01–2022-12) of INRIX travel time data
- I-35 Northbound with 12 months (2022-01–2022-12) of INRIX travel time data
- I-35 Southbound with 12 months (2022-01–2022-12) of INRIX travel time data

10.2.4. Phase I Models

The output of the RNN training process is a set of models that take in datapoints for the current timestep and output a prediction of segment-level travel times at a given timestep in the future. These models are iteratively improved by evaluating the change in prediction MSE for the validation dataset as a function of the input training data. The MSE acts as the objective function of the optimization problem which we seek to minimize, but we aim to avoid overfitting (the process of training a model to predict the input training data alone, without good generalizability). To do so, we follow the standard procedure of withholding the validation and testing datasets discussed earlier and penalize the RNN parameter optimizer for models that predict these withheld datasets poorly—this avoids letting the RNN train over the full dataset so as to enforce the aim of developing a model that generalizes to all data rather than the training dataset alone.

We further evaluate the predictive strength of our models using the coefficient of determination (R^2) and mean absolute error (MAE) metrics.

In general, we see these error metrics degrade as we forecast further into the future, but near-term results show promise. While they are not yet strong in their predictive power (e.g., R^2 does not exceed 0.6 for the studied corridors), we believe further training and model refinement may improve performance.

Figure 82 presents the R^2 as a function of the number of timesteps into the future for which predictions are generated. In general, results show a trend across all corridors in which predictive power drops considerably as we predict further and further into the future. While this is not unexpected, the predictive power of the models falls significantly after two timesteps. Potential issues for the observed fit include:

Limitations in input data. Our models were trained using data for year 2021, and traveler behavior may have changed considerably throughout the year due to the impacts of the COVID-19 pandemic, leading to poorer results (given that we are not including data which could predict changes in behavior in response to local outbreaks). Additionally, annual seasonality has not been explicitly considered. Future work may address this by adding more years' data to the training dataset and/or providing an input variable indicating the week or day of the year at which the input datapoint was collected.

RNN design. The training experiments described in this document used the same RNN design used in our initial protype for I-35 through Austin. While in the Austin case the single-hidden-layer RNN led to very promising results, adding hidden layers is often better suited to approximating complex nonlinearities such as weekly or seasonal patterns in input data. Future work may investigate the benefits of adding both additional records (i.e., adding more than one year of data) and/or reshaping the structure of the RNN, as both would be expected to improve model generalizability.

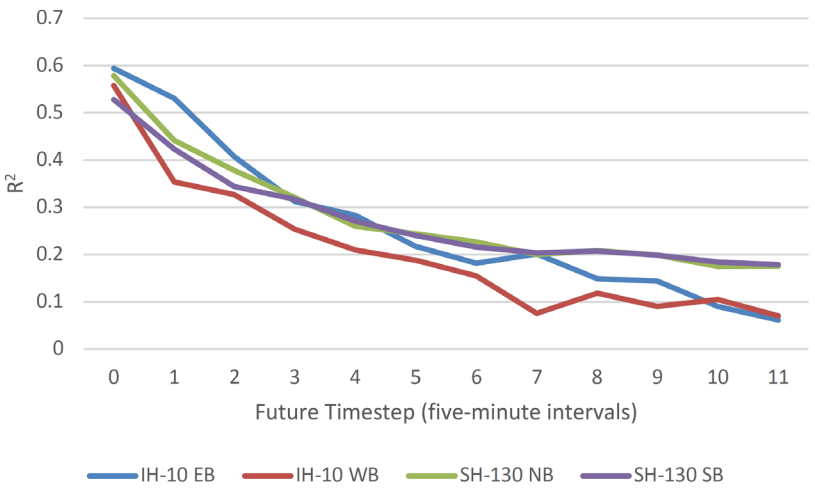


Figure 82: R^2 values for models trained on a full year's data.

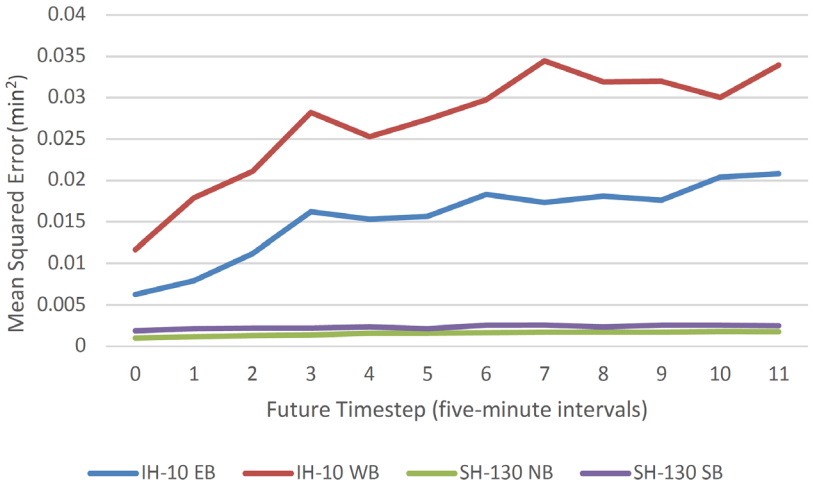


Figure 83: Mean squared error for models trained on a full year's data.

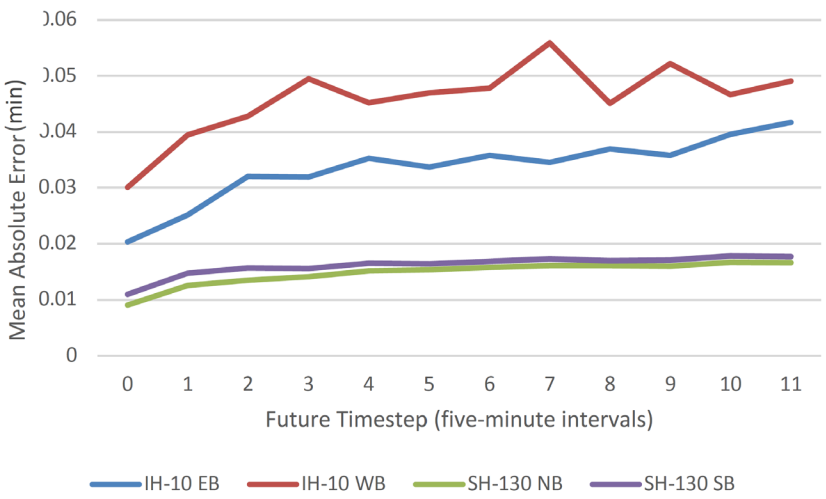


Figure 84: Mean absolute error for models trained on a full year's data.

R^2 values on the testing dataset were in the same range as those observed in the training dataset, indicating relatively good generalizability of the model and a lack of overfitting.

Figure 85 provides an additional illustration of the model performance on the I-10 eastbound corridor. In this figure, read left to right then top to bottom, each graph shows the true segment travel times (X axis) compared to the predictions from the models (Y axis) for the testing dataset. Each subsequent graph is forecasting travel times a further five-minute timestep into the future. What we generally see is that the model has difficulty predicting conditions of high travel times, most likely caused by non-recurring congestion such as is caused by accidents. We also see a tendency to underestimate travel times, possibly due to the lack of volume data in these models to indicate how susceptible each segment is to being oversaturated.

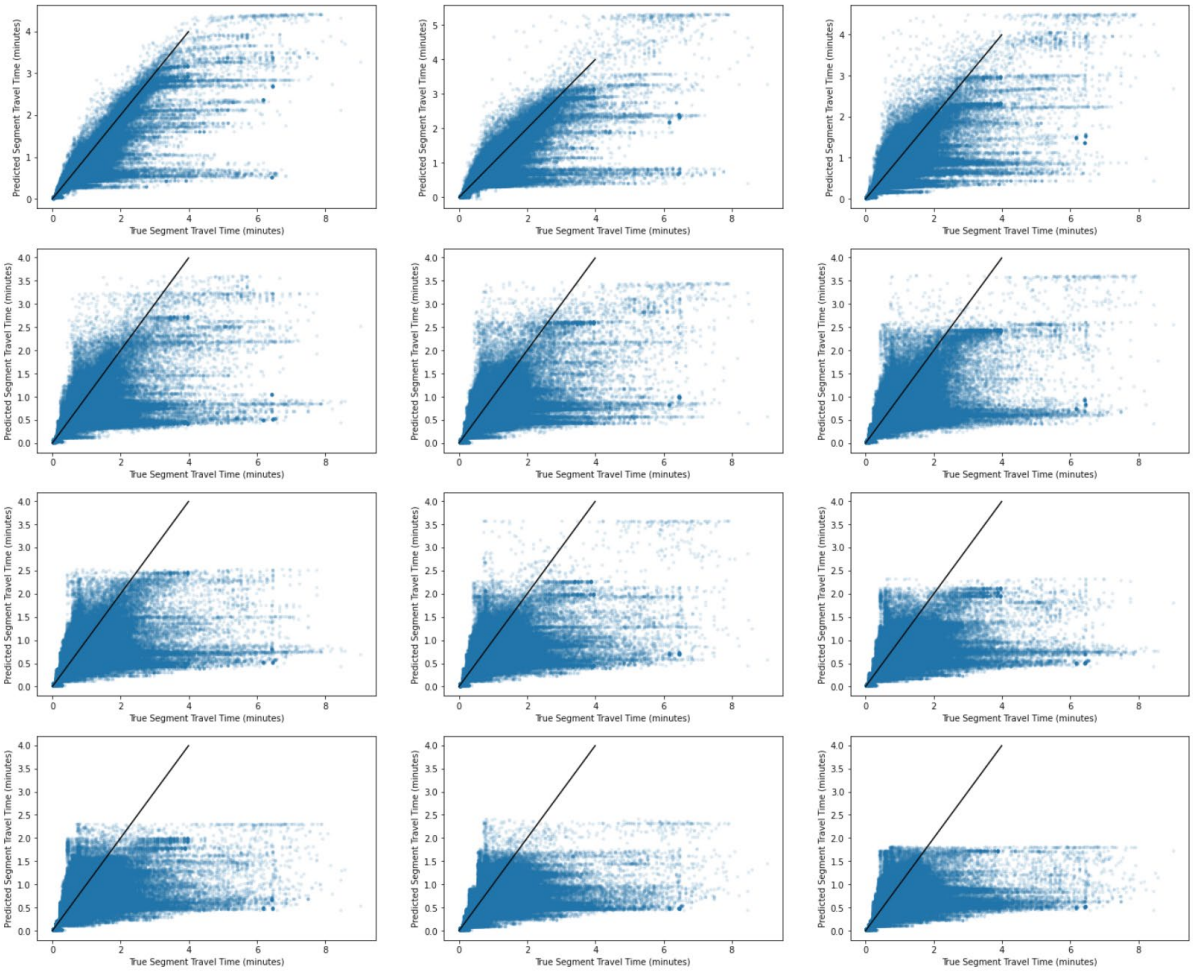


Figure 85: Correlation plot of testing data for I-10 eastbound forecasts.

Figure 86 through Figure 88 compare the performance of models trained with and without volume data using R^2, MSE, and MAE. All other experimental parameters being held constant, we see somewhat similar performance between the models in terms of predictive power. Results show a decline in predictive accuracy as we look further ahead into the future for both models. We also see that, in most cases, further into the future, the models with volume and occupancy data show an improvement over the model without the added data.

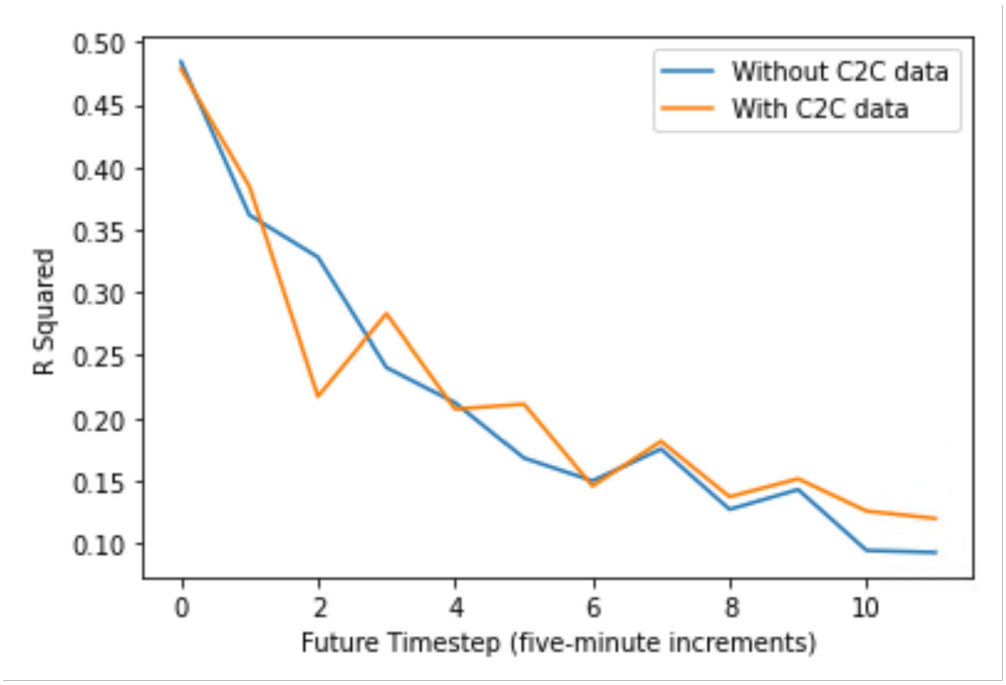


Figure 86: Testing R^2 values for models with (orange) and without (blue) C2C data.

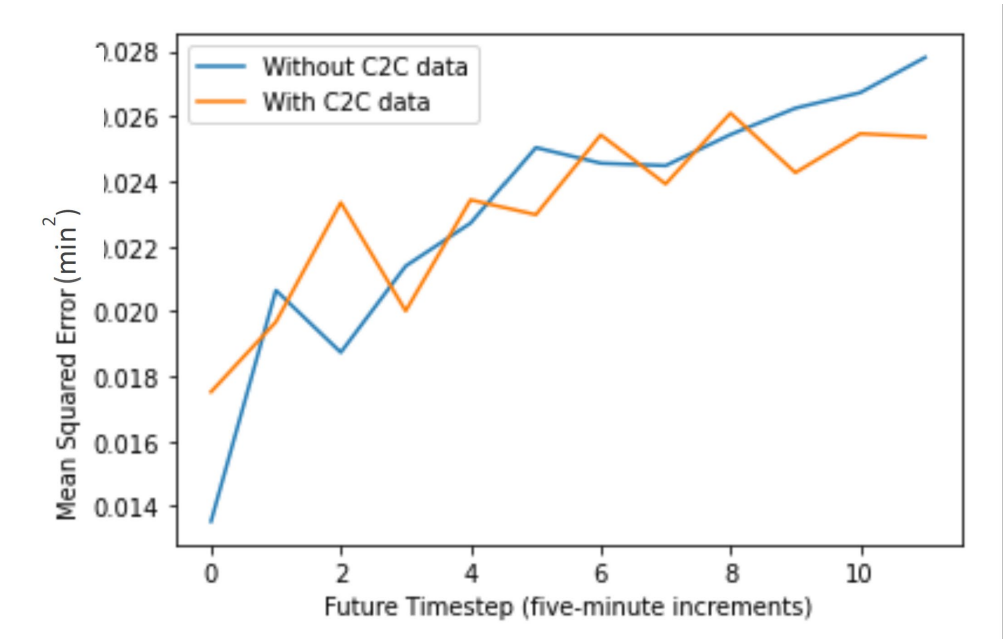


Figure 87: Figure 10: Testing MSE values for models with (orange) and without (blue) C2C data.

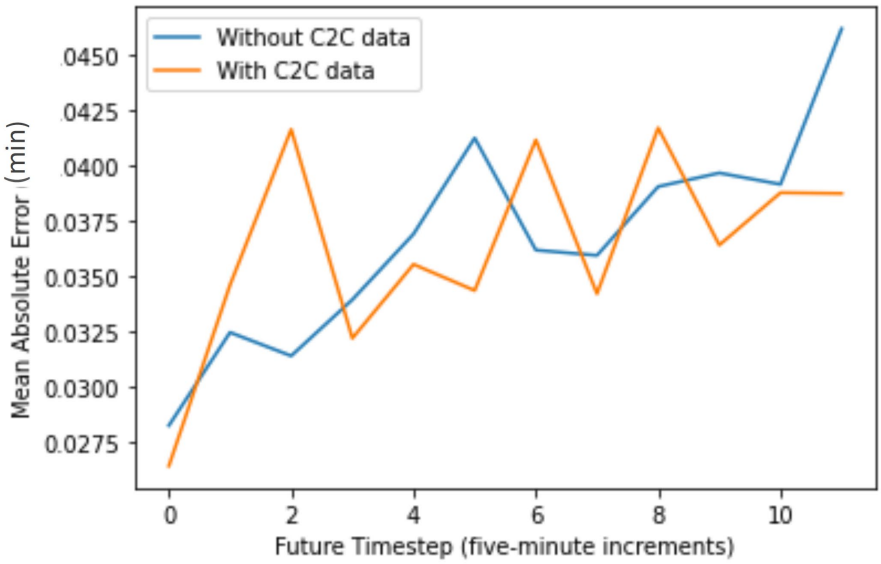


Figure 88: Testing MAE values for models with (orange) and without (blue) C2C data.

In the nearer-term predictions, inclusion of volume and occupancy do not necessarily improve predictions. We suspect this may be caused by a combination of factors. First, while the model is being trained using additional data sources, no compensation was made in model or training design to account for this data. Generally, it is advisable to increase the RNN size and/or prolong training when additional data dimensions are provided. As a result, the model without C2C data may have been able to discern more clearly the importance of the individual data points it received due to not expending computation time determining the importance of the added points as well. Second, in this experiment, we train the model only on data ranging from December 2021 to May 2021, due to limited data availability from the Lonestar C2C data source. This in turn may cause a severe impact on predictive power due to the seasonality of traveler behavior and lack of sufficient examples to train the model.

Initial models developed on the secondary corridors allowed researchers to refine the code base used for model-developed tools that may be more easily extended to the analysis of new sites and datasets. The results obtained using a simple RNN design are encouraging, given the low MAE values. However, the predictive power of the models is improved in later iterations of the training process (Phase II).

10.2.5. Phase II Models

The tests on previous ML models demonstrated value and built a robust environment to train and test new models. Though some initial models did not show significant benefits over current practice, their development and testing suggested avenues for improvement. In addition,

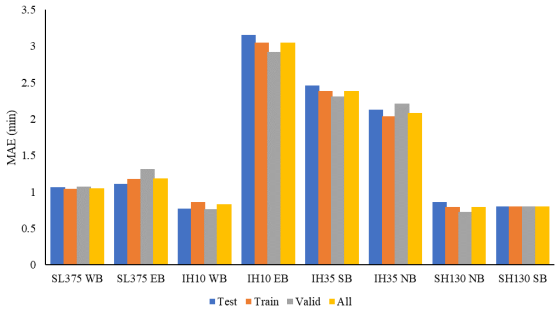
previous models demonstrated that models trained on the segment level performed better than those on the route level. Building on this framework, a new set of ML models was developed using several different spatial aggregation methods. These models are analyzed in depth in this section and their value for practice is explained.

The automated workflows developed in this project allow for the travel times on small paths to be predicted and then aggregated up to the route level. For each route, three levels of spatial aggregation were estimated. The single segments model estimates the travel time on each INRIX segment separately (with lengths less than a mile) and then aggregates these estimates up to an estimate of the total corridor travel time. In previous sections, we presented low MAE on the I-10 EB corridor, while this section extends the findings to additional corridors. The pair-segments model does the same, but predicts travel times for paths consisting of two consecutive segments. Finally, the route-level model utilizes all the data for the corridor and uses a single ML model to make a forecast. We will demonstrate that even though the models with less spatial aggregation (single segments and pair segments) are more computationally expensive (since many ML models need to be run), they have much better performance than a single ML model predicting travel times for the entire corridor. Finally, another set of models (referred to as congestion models) was developed where subsets of each corridor were identified and expected to have similar patterns of congestion propagation. These slightly longer paths were developed to be an intermediate aggregation option. These models were developed and run only for the I-35 and SL 375 corridors.

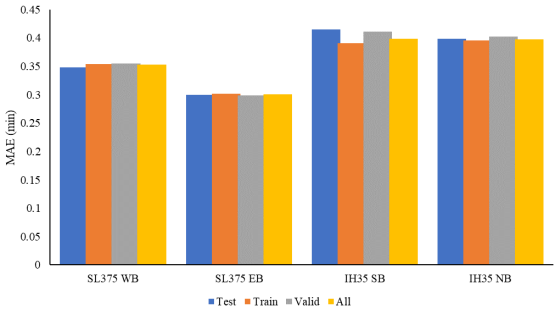
10.2.5.1. Training Results

This section utilizes the MAE and R^2 as the primary evaluation tools. Researchers examined the other metrics developed in previous sections, including the MAE, R^2, MSLE, MDAE, and MAPE. These other metrics followed the trends of the MAE and R^2, so they are omitted for concision. We note that for shorter segments it is possible for the ML model to predict a negative travel time. Future research should develop optimization methods to ensure this does not occur. However, when predicting travel times on an entire corridor (or segments longer than one mile) this is not an issue. For the purposes of this study, negative travel times are allowed but are set to zero for the MSLE calculation (the negative values are included in the evaluation of the route level travel times).

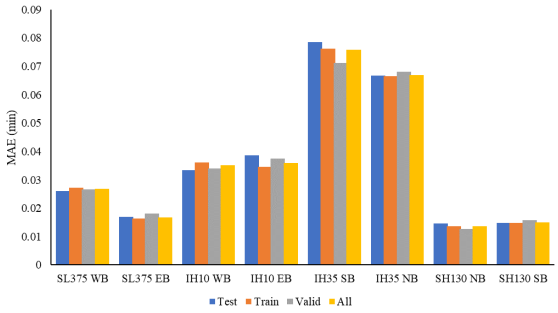
Figures 89 and 90 depicts the travel time prediction MAE and R^2, respectively, for step 0 (departure time). For disaggregate modes, the reported value is the average across all considered spatial units (e.g., segments). The training, validation, and testing data match well across all models and corridors. The route-level model error is significantly larger because the corridor-level travel time is longer. Similarly, longer routes/paths are expected to exhibit larger errors.



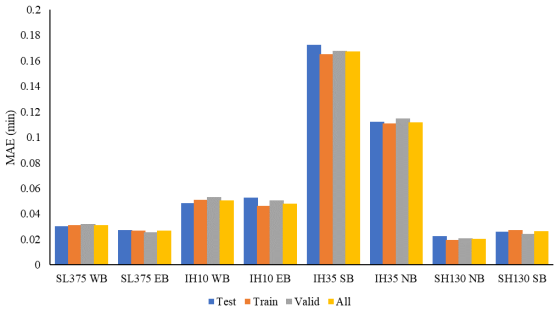
a) Route-level models.



b) Congestion paths models.



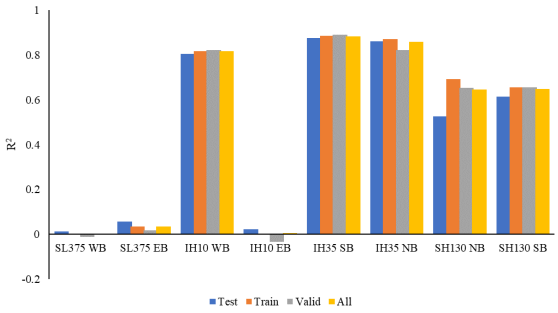
c) Segment-level models.



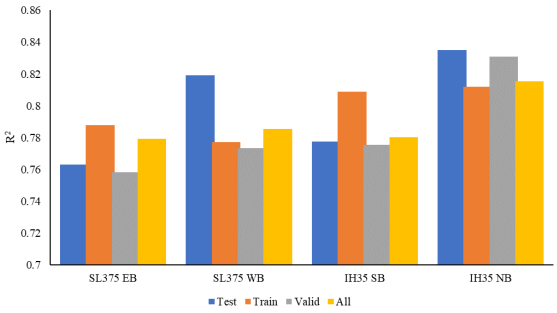
d) Segment pair level models.

Figure 89: Average MAE by route for predictions zero timesteps ahead (time t).

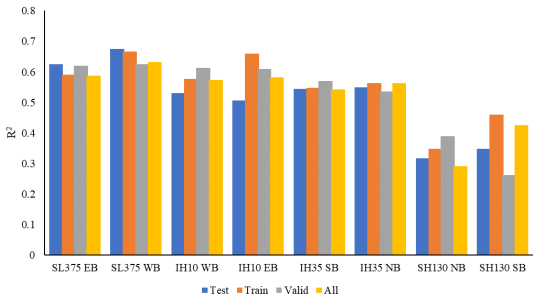
Figures 91 and 92 examine the performance further into the future on the I-35 SB routes. These figures depict the MAE and demonstrate increasingly poor performance further into the future. This trend is most pronounced for the congestion paths model, where there is a very sharp increase in MAE for the route-level model after 25 minutes. The other two models are more stable in time and have substantially lower MAE. Again, this is in part due to the shorter length of the paths. Unfortunately, even accounting for this, more aggregate models still do not perform as well.



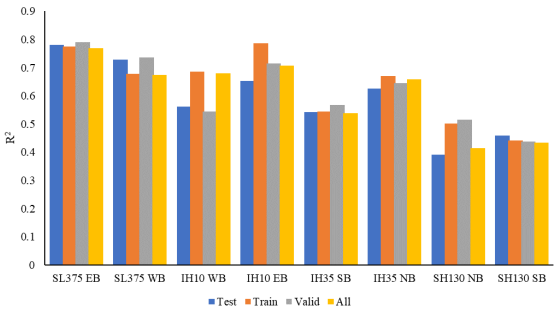
a) Route-level models.



b) Congestion paths models.

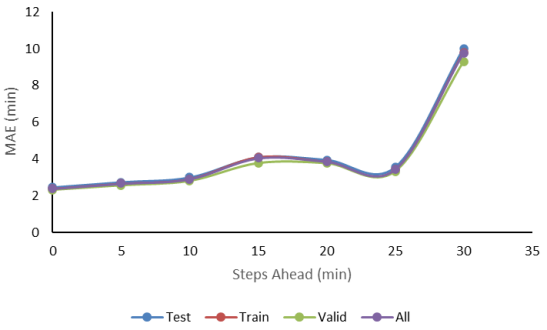


c) Segment-level models.

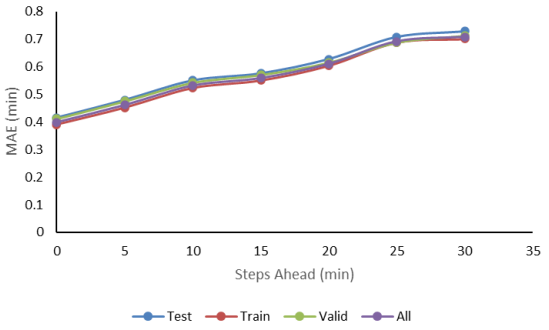


d) Segment pair level models.

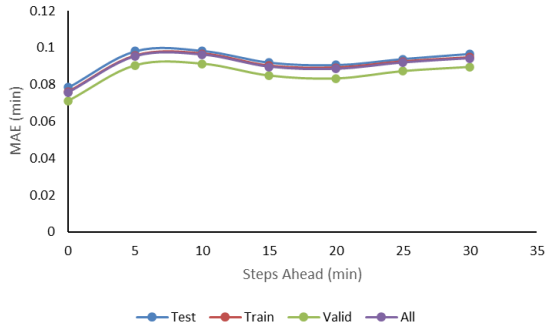
Figure 90: R^2 by route for predictions zero timesteps ahead (time t).



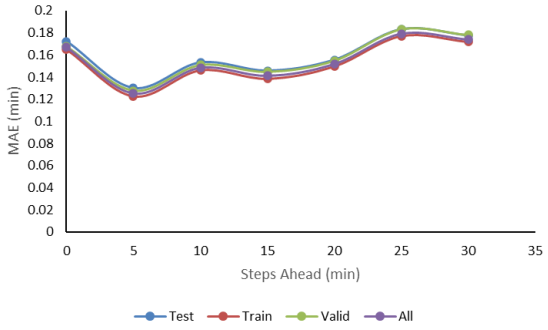
a) Route-level model.



b) Congestion paths model.

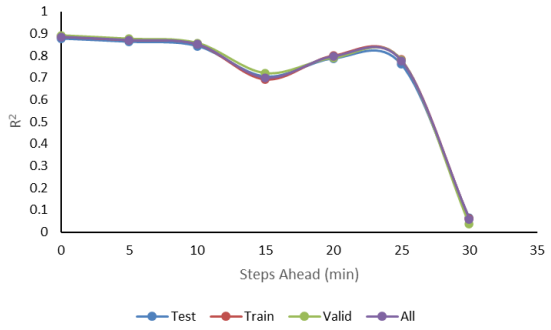


c) *Single segment model*

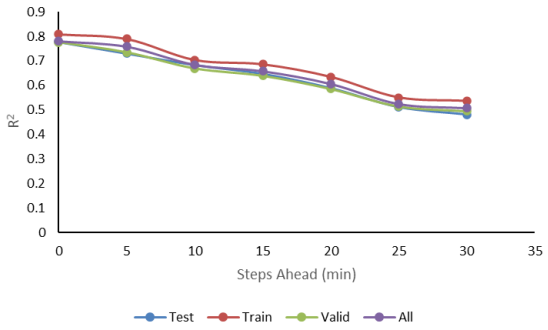


d) *Pair segment model.*

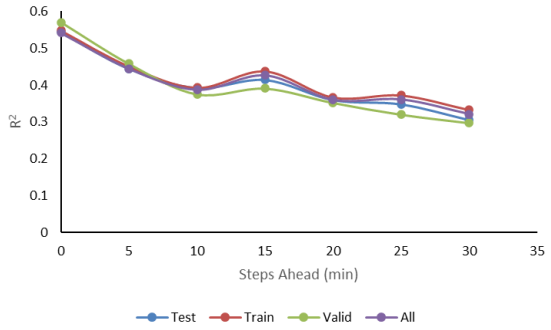
Figure 91: MAE (min) for timesteps t+0 to t+30 on I-35 SB.

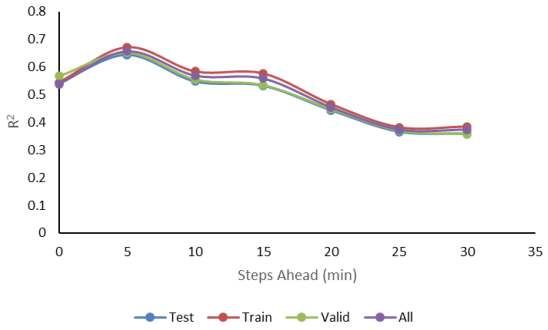


a) *Route-level model.*



b) *Congestion paths model.*





c) *Single segment model* d) *Pair segment model.*

Figure 92: R^2 for timesteps t+0 to t+30 on I-35 SB.

10.3. Continuous Evaluation Framework and Results

The performance of the machine-learning-based dynamic travel-time models will be compared against two simpler models: a dynamic model that forecasts downstream segment travel times using weakly mean values, and a static model that implicitly assumes that travel times in downstream segments will not change from the previous timestep (which is consistent with current practice).

This section documents the measures proposed by CTR to evaluate deployed travel time prediction models in real time and periodically. The goal of real-time evaluation is to support decision-makers in their use of the model outputs, providing a means to decide if (and which) information should be provided to drivers, and whether it may be appropriate to take proactive measures to alleviate forecasted traffic conditions. Periodic model evaluations are intended to assess long-term model performance, guide model improvement and refinement tasks (including any need for re-training) and support the estimation of the value of trained models.

Our performance evaluation approach uses archived real-time travel time predictions from deployed models, which are stored in a relational database hosted by CTR. The same database is used to archive realized travel times, computed using INRIX data and a dynamic approach further described below, and two "baseline" travel times that represent current practice and average conditions.

Average results on a testing dataset that consists of 71 days (42 days for C2C-enabled models) that were not used in model training are used to exemplify the proposed metrics for our INRIX-only models, while those incorporating C2C data withheld 42 days' records. CTR has also deployed workflows to visualize some of these metrics in real time through the web application and to download periodic reports (Product 6). The evaluation framework has been refined and extended based on TxDOT's feedback.

10.3.1. Performance Evaluation Approach

In this effort the performance of machine learning models for the prediction of expected travel times along selected origin-destination paths (or corridors) is based on comparing model forecasts to realized travel times at 5-minute timesteps. In the context of this project, realized travel times are obtained dynamically using INRIX data and the workflow presented in Figure 93. Dynamic corridor travel times for start time *t* are obtained by traversing the segments of a corridor and adding their individual travel times, with the first segment in the sequence

evaluated at *t* and the remaining segments evaluated at the observation closest to the time at which a traversing vehicle would have entered it. These segment-level travel times are summed to provide a path travel time indexed by the time at which the traversal began. Incomplete traversals (those whose travelers have not yet traversed the complete sequence of segments) are left undefined until the traversal can be completed.

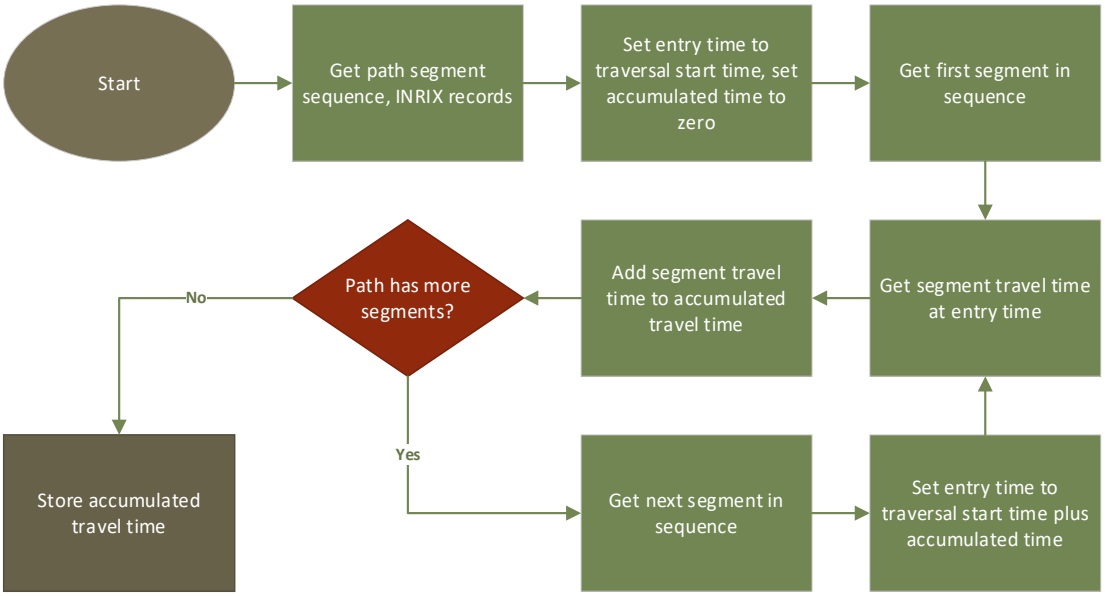


Figure 93: Path travel time calculation algorithm.

We also compare the experienced travel times estimated using machine learning models to two baselines: the instantaneous travel time and the mean travel time at that time of week. The instantaneous travel time is used as the current practice by most agencies and assumes that the travel time for each roadway segment will remain the same as it was when measured at the beginning of a roadway traversal (this is referred to elsewhere in the report as the naïve model). The mean travel time, by contrast, calculates the average travel time for the full path across past records and does not consider real-time data.

Additional data sources may be used in the future to supplement the proposed evaluation approach. Current limitations are mostly related to the computation of actual experienced travel times and include the potential presence of biases in INRIX data that would not be captured by our method (e.g., systematic over or under estimation of segment-level travel times). Realized travel times used in this effort may also be affected by the use of spatially aggregated data for their calculation. Using aggregate segment-level data to reconstruct the travel time experienced by individual travelers may fail to capture delays that are specific to a particular type of movement, such merging into/out of an access ramp, which in some cases may lead to biased estimates of experienced travel times for specific origin-destination pairs.

10.3.2. Lookback Windows

The proposed performance evaluation metrics are computed using data that is collected/generated in real time when models are deployed. Figure 94 provides a schematic description of the data structures used to archive relevant information in a relational database for this project. CTR has identified a series of lookback windows across which to evaluate deployed models, consisting of the following time periods: one hour, three hours, six hours, 12 hours, one day, and one week, with each window ending with the most recently completed traversal travel time and its accompanying predicted value. We also calculate our summary metrics for the model's performance since its inception and provide monthly performance reports. In providing a variety of lookback windows, we can determine any changes in performance in response to recent changes in corridor traffic, whether related to incidents or longer-term changes in roadway characteristics. We can also provide insight on the quality of the model outputs for a specific time step to support decision making.

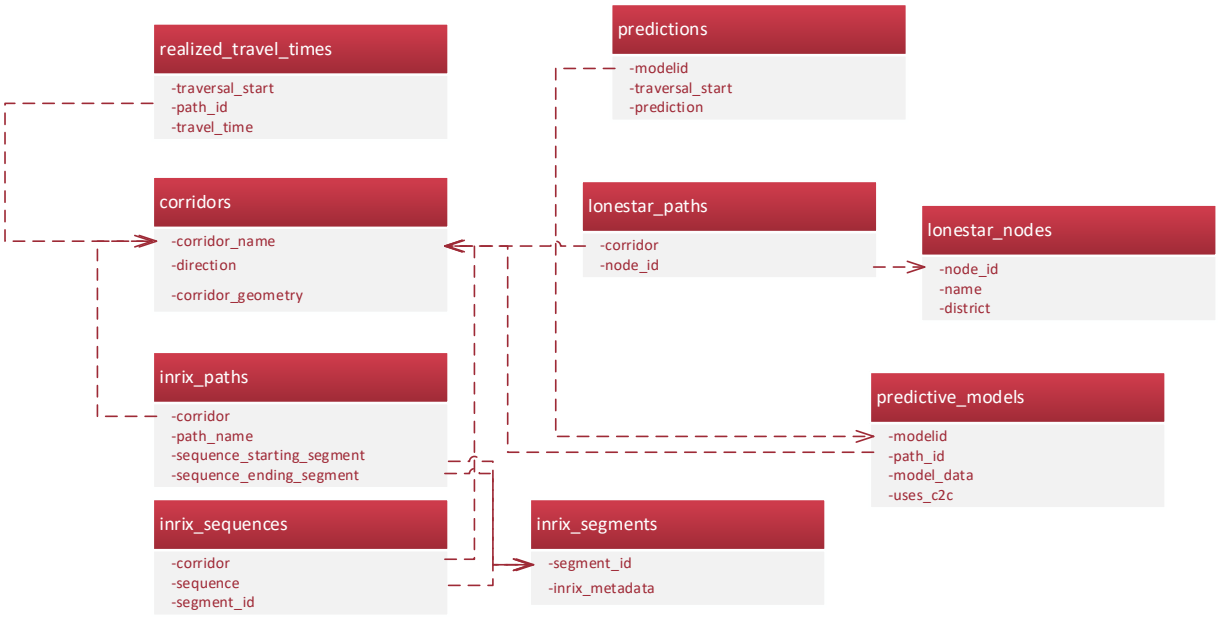


Figure 94: Simplified database schema.

10.3.3. Metrics

As a means to evaluate the performance of our models along multiple dimensions, we calculate a series of metrics for each corridor's lookback windows. These each provide distinct insights into the performance of the model, with attributes that are not incorporated into the other metrics. In this section, we define and discuss each of the metrics we evaluate. Throughout this section, we denote the predicted travel times as, the observed true travel times as, and the mean of all predictions as.

The metrics discussed below will be used in three ways:

- To provide insights into the quality of travel time predictions in real-time to support decision makers in their use of the data.
- To evaluate and refine long-term model performance by conducting analyses that consider variability across hours of the day, days of the week, and between typical/atypical traffic conditions.
- To identify deterioration/improvement of the model performance over time, understand when re-training may be necessary, and assess the benefits of new/re-trained models.

10.3.3.1. Mean Squared Error

For a series of estimated values and their corresponding true values, the mean squared error (MSE) statistic takes the squared errors across all data points and averages them. MSE acts as a means to evaluate *risk* associated with our models, as the values are strictly non-negative and therefore represent the quality of the model as a measure of the Euclidean distance from the true values. The units of the metric are the squared units of the data points (in our case, travel time in minutes). The mathematical definition of MSE for a set of data points is shown below:

$$MSE = \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{n}$$

In our experiments, we select this metric as the performance evaluation metric during model training so the models learn how to best predict data points in order to minimize the MSE.

10.3.3.2. Mean Absolute Error

Mean absolute error (MAE) is similar in nature to the MSE metric but does not square the error terms of each prediction. While this metric provides the advantage of providing error in units identical to the data points being evaluated, it creates a shortcoming in that it does not provide as strong a disincentive for estimates that are far from the true values as MSE. The formulation of MAE is shown below:

$$MAE = \frac{\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right|}{n}$$

10.3.3.3. Coefficient of Determination – R^2

The coefficient of determination for a predicted and true data set is a goodness of fit metric and defines the proportion of variation in the true values explainable by the predictive model. It can be visualized as the slope of a regression line in which the dependent variables are the true values, and the independent variables are the predicted values, with values typically falling between 0 and 1. An R^2 value of 1 indicates that the model perfectly predicts the true values,

while a value of 0 represents a distinct lack of predictive capabilities—that is, the model is completely unable to predict the true values relative to the worst-possible least-squares predictor (i.e., simply using the mean of the estimates, as each data point estimate). We use the following definition of R^2 in our models:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

10.3.3.4. Mean Squared Log Error

The mean squared log error (MSLE) is similar to the MSE metric but is designed to penalize a low estimate more than a prediction that is too high. This can be useful in circumstances where providing a conservatively high prediction is useful. The definition we use for MSLE is:

$$MSLE = \frac{\sum_{i=1}^{n}\left(\ln(1 + Y_i) - \ln\left(1 + \hat{Y}_i\right)\right)^2}{n}$$

10.3.3.5. Median Absolute Error

The median absolute error (MdAE) metric is similar to the MAE metric but is more robust to outliers. By incorporating a median function, the magnitude of each outlier is strongly diminished in significance. The mathematical formula is not easily expressible in closed form, however, due to the median function, taking the central value from the ordered set of absolute errors, i.e., $\left\{\left|Y_i - \hat{Y}_i\right| \forall\, i\right\}$.

10.3.3.6. Mean Absolute Percent Error

Much like the MAE metric, mean absolute percent error (MAPE) is sensitive to the absolute error of predictions, but specifically considers the size of the error relative to the mean value. This allows for direct comparison of models that may have different scales of their target variables (for example, 2 corridors with different free-flow travel times). The definition of MAPE is:

$$\frac{\sum_{i=1}^{n}\frac{Y_i - \hat{Y}_i}{|Y_i|}}{n}$$

As a note, when is zero, the value must be treated as an arbitrarily small positive number. Given the application of the models we are using, this is generally not relevant, as travel time predictions ideally should never be zero.

10.3.4. Sample of Performance Metric Computation

To illustrate the use of the metrics proposed in Section 10.3.3, we have computed their value using a 2021 dataset that was withheld from the model during training. We also illustrate their value when computed for a single point in time. Future work will consider the performance evolution over longer time periods (the models evaluated in this section were only run for two weeks). In the analyses below we compare the performance of our trained machine learning models to two baselines:

- A naïve model, which assumes that current travel times will not change as travelers traverse the corridor. This model is typically used in practice by most agencies and predicts the experienced travel time to be the sum of current travel times on all segments.
- The typical travel time value for the specific timestep. Typical travel times for each 5-minute time step and day of the week are computed using the training dataset.

10.3.4.1. Evaluation of average model performance

Table 18 presents the results for six trained models initially deployed, which were retrained as of September 2021. Results suggest that trained models can forecast experienced travel times on the selected corridors, with the C2C-enabled models providing the most explanatory capabilities, as shown by the highest R^2 values. The MAE suggests that model predictions are, on average, within one minute of the true values for most models and are a good indicator of the real-world promise of these models. In addition, all trained models exhibit MAPE values under 3 percent and a strong predictive power based on the R^2 values, which exceed 0.75. The latter suggests that trained models are expected to provide significantly better estimates of experienced travel times than a model which simply uses the mean observed value.

Table 18: Testing metrics for the trained models

Model	MSE	MAE	R^2	MSLE	MdAE	MAPE
I-10 EB w/ C2C	1.163	0.765	0.925	0.001	0.570	0.025
I-10 WB w/ C2C	3.400	1.024	0.893	0.002	0.449	0.029
I-10 EB w/o C2C	1.405	0.635	0.799	0.001	0.400	0.021
I-10 WB w/o C2C	1.418	0.675	0.859	0.001	0.378	0.022
SH 130 NB	1.961	0.693	0.687	0.001	0.469	0.014
SH 130 SB	1.530	0.685	0.761	0.000	0.446	0.013

10.3.4.2. Assessment of the quality of specific travel time estimates

CTR deployed the trained models to predict the corridor travel time at 5-minute intervals, and a separate module to evaluate their performance for each of the lookback windows defined in

Section 10.3.3. Table 19 presents an example of the metrics for the southbound SH 130 corridor across all lookback windows from a single point in time corresponding to an evening rush hour.

Table 19: Metrics for the Southbound SH 130 model

Timesteps	Hours	MSE	MAE	R^2	MSLE	MdAE	MAPE
12	1	2.123	1.392	-5.613	0.001	1.339	0.028
36	3	0.870	0.690	-0.401	0.000	0.450	0.014
72	6	0.698	0.642	-0.476	0.000	0.466	0.013
144	12	0.985	0.684	0.607	0.000	0.466	0.014
288	24	1.841	0.993	0.620	0.001	0.691	0.020
2000	166.6667	5.654	1.222	0.467	0.002	0.756	0.023
3655	304.5833	3.975	1.127	0.458	0.001	0.741	0.022

The goal of computing these metrics at each time step is to assess the quality of a specific travel time prediction, and to support decision makers in their use of the data. Table 19 suggests that the model performance over the last hour is significantly worse than the average performance observed during training. Further, Table 20 suggests that a naïve model may provide a better estimate at the considered point in time. The average performance of the RNN model over the preceding 12 hours is close to the performance observed during training, but the 24-hour performance is again below expectations. The latter, in combination with the observed performance over longer lookback windows, is an indication of potential issues in a deployed model. CTR has used the insights from this analysis to identify a problem in the models initially deployed, which is addressed in future model refinements.

Table 20: Metrics for the Southbound SH 130 naïve prediction method

Timesteps	Hours	MSE	MAE	R^2	MSLE	MdAE	MAPE
12	1	0.1182	0.2832	-0.2491	0.0001	0.297	0.0059
36	3	0.2492	0.3998	-0.1415	0.0001	0.3575	0.0084
72	6	0.4954	0.5051	0.4436	0.0002	0.3975	0.0105
144	12	0.7446	0.6053	0.6586	0.0003	0.4125	0.0123
288	24	1.3537	0.8248	0.3181	0.0005	0.5535	0.0167
2000	166.6667	4.6295	1.0837	0.5569	0.0013	0.565	0.0207
125633	10469.42	2.3367	0.8384	0.7861	0.0007	0.52	0.0162

A decision maker faced with these results may choose to display only the naïve prediction, the model forecast prediction, or some combination of the two such as a range. During the remainder of this project, CTR will monitor model performance using the proposed metrics and develop guidelines to support the use of the data.

10.3.4.3. Discussion

This section described a set of performance metrics and a framework to compute them in real time and over extended time periods. The proposed metrics may be used to assess the quality of model predictions in real time, to evaluate and refine long-term model performance, and to identify changes in the model performance over time that may suggest the need for re-training.

CTR demonstrated the use of the proposed metrics to evaluate deployed models using a testing dataset. This effort suggests that trained models have good predictive power and lead to experienced travel time estimates that are within one minute (and 3 percent) of measured travel times. CTR also illustrated how to use real-time metric values to support the use of travel-time predictions by decision makers. The framework can provide indication of whether a model should be used to forecast travel times rather than current state-of-the-art practices. If predictions are found to be accurate, they may aid decision makers in determining whether to take proactive measures to alleviate anticipated congestion. Proposed metrics can also be used to identify potential issues in a model deployment and the need for further training.

CTR has developed a software application that provides the proposed metrics for a model across a set of lookback windows, starting with the most recently completed traversal and extending backward across several spans of time. CTR has also extended the web-based application to compute and visualize real-time and aggregated metrics of performance. The research team has continued to monitor these metrics throughout the project to improve the performance of deployed models and provide guidelines to TxDOT in the use of the generated data.

Preliminary results from this framework helped CTR notice an issue in the models initially deployed. The team has identified approaches to address observed discrepancies between the model performance on the testing dataset and the performance observed during real-time estimation.

10.3.5. Real Time Evaluation of Phase I Models

This section describes the selected approach to evaluate the real-time performance of the machine learning models deployed by CTR for the estimation of experienced travel times on selected freeway and highway corridors in Texas. The results presented here include updated real-time performance metrics for models that have been running since November 2022 and a description of new models deployed in March 2022, including a summary of their performance on the testing dataset when available.

The following sections describe the types of models tested by CTR in this section, the corridors analyzed and the selected metrics for real-time performance evaluation.

10.3.5.1. Overview of models under testing

A variety of dimensions have been explored through which we can modify models to evaluate their efficacy. We first compared two separate corridors (SH 130 in Austin and I-10 in El Paso) for their relative predictive power, with comparable results in both. This is promising, as it shows that the model setup is transferrable to different corridors with different characteristics.

We further analyzed the effect of providing C2C volume, speed, and occupancy data to the models for I-10. While the models incorporating the additional data performed better in training, they have underperformed relative to the simpler models while testing in real time. This may be due to changes in sensor availability or accuracy or may reflect a change in travel behavior in the El Paso area since the initial training occurred. Future analysis aims to determine a path forward to improve the usage of C2C data in these models, as this theoretically should provide better results given relatively accurate input data.

Finally, in conjunction with the future forecasting models, we explored the impact of subdividing corridors into components (hereafter referred to as paths) which are predicted independently. These models require, for the purpose of constructing an overall corridor traversal time, future forecast models for each component path, as the forecast travel time for a component path may change between when a corridor traversal starts and when a traveler reaches the start of a component path.

Next, we evaluated models' predictive power looking further into the future than an immediate forecast. That is, while initial models predicted corridor traversal time if traversal was begun at the time of the forecast, the new models predicted later. Specifically, we trained models to predict up to an hour into the future relative to the latest data point. Unsurprisingly, these models show a generally worse predictive capability as they look further ahead, with immediate forecasts generating R^2 values in the vicinity of 0.8 and hour-later forecasts closer to 0.25. Despite this, we still see the later forecasts holding MAE to within 1.5 minutes of the true travel time along the corridor.

For future forecasting and multi-path corridor models, we were predictably met with an increase in time and memory required for training these models, so much so that the models were unable to be trained simultaneously due to the diseconomies of these model structures. To remedy this, we trained each future forecast separately, as these structures are independent, though the forecast for each component path for a given time occurs within the same model.

An area for future study may involve switching these two dimensions—having a model that predicts, for a given component path, the travel time at each timestep in the future, and multiple such models being stitched together to predict for a full corridor. Intuition would tell

us that this approach may be less effective, as traffic is somewhat Markovian in nature, depending less on the state of traffic in the distant past but heavily dependent on the immediate past. By comparison, spatial relationships in traffic are well studied, and any attempt to split corridors may introduce arbitrary boundaries near which predictive power is significantly different than the rest of the path.

Considering the increase in needed resources for training models that forecast for multiple paths, future work will explore taking this approach to its logical extreme—modeling the travel time on each INRIX segment in the component corridor. This will provide the highest-resolution forecasts possible but will require even more resources to train.

10.3.5.2. Analyzed corridors and deployed models

We have focused on four corridors in our preliminary experiments: SH 130 and I-35 through the Austin metro area and I-10 and SL 375 in the El Paso area. The table below details the routes tested in experiments to date.

Table 21: Route testing summary.

Corridor	Start	End	Length	Path Start	Path End	Length
SH 130 NB	I-35 @ SH 45	I-35 @ SH 130	55 mi			
SH 130 SB	I-35 @ SH. 130	I-35 @ SH 45	55 mi			
I-10 EB	SL 375 @ I-10	SL 375 @ I-10	32 mi			
I-10 WB	SL 375 @ I-10	SL 375 @ I-10	32 mi			
I-35 NB	SH 130 @ I-35	Riverside @ I-35	33 mi	Riverside	US 183	7 mi
				US 183	FM 734	8 mi
				FM 734	SH 45	8 mi
				SH 45	US 79	3 mi
				US 79	SH 130	12 mi
I-35 SB	Riverside @ I-35	SH 130 @ I-35	33 mi	SH 130	US 79	12 mi
				US 79	SH 45	3 mi
				SH 45	FM 734	8 mi
				FM 734	US 183	8 mi
				US 183	Riverside	7 mi

For the above routes, we developed a variety of models that are summarized in the table below:

Table 22: Model testing summary.

Corridor	Number of Paths	Uses C2C?	Training MAE	Training MSE	Training MAPE	Deployed
SH 130 NB	1	No	0.83	2.66	1.62%	Sep 2022
SH 130 SB	1	No	0.69	1.57	1.36%	Sep 2022
I-10 EB	1	Yes	0.13	0.20	0.42%	Sep 2022
I-10 WB	1	Yes	0.12	0.36	0.36%	Sep 2022
I-10 EB	1	No	2.19	9.63	7.30%	Sep 2022
I-10 WB	1	No	1.75	9.24	5.50%	Sep 2022
I-35 NB	5	No	N/A	N/A	N/A	No
I-35 SB	5	No	N/A	N/A	N/A	No

10.3.5.3. Evaluation Approach

The evaluation approach proposed in this document is intended to evaluate three major aspects of the real-time performance of the deployed ML models:

Consistency with the model performance observed during training, testing, and validation.
Magnitude of the error by time of day.
Relative performance when compared to non-predictive methos to estimate experienced travel time (baseline models described in previous TMs).

We use the metrics described in previous sections to measure the performance of deployed models by comparing estimated and realized values of the experienced travel time. Given the fine temporal resolution at which travel time estimates are produced it is important to consider different strategies for aggregation to derive meaningful insights. We propose three analysis approaches, which may be further extended and refined:

Aggregate performance: This approach computes the average performance using the entire training, testing, validation, and real-time data sets. This type of analysis will be used to understand whether the models are performing as expected based on the training process. Discrepancies may lead to the identification of issues such as overfitting, live data errors, and potential deployment errors/inconsistencies. While we use the entire datasets described earlier in this analysis, results are also provided for meaningful subsets of the data. The following three categories are currently being considered.

Average results: computed using entire dataset (24 hours of data for each analyzed day)
AM/PM peak period results: computed using data between 6 AM and 9 AM (AM) and 4 PM and 7 PM (PM)

Typical/atypical traffic results: distinguish between data points that correspond to typical and atypical traffic conditions (traffic conditions are defined based on realized travel times as defined in previous TMs)

Performance by hour: This analysis is conducted using the real-time data set, and it consists of comparing ML models to baseline estimation approaches. The approach is expected to aid in the quantification of the value of ML models, and to provide a more detailed understanding of performance pattern. The latter can support further model refinements and the development of guidelines for their use.

Disaggregate Performance: This approach consists of a visualization of the error of individual datapoints in a heatmap. The results provide a detailed picture of ML and baseline methods by time of day and day of week and will support the identification of error patterns that may be corrected by adjusting models' specification and/or training processes. This analysis will also inform the development of guidelines for the use of the final models.

10.3.5.4. Example of analysis

CTR analyzed model performance using the approach described in the previous section for all models deployed in November. The analysis uses training, testing, and validation data along with real-time data. The latter was collected after fixing all known issues in the real-time data pipeline (November 17, 2022). Data is available for two bi-directional corridors, SH 130 in Austin and I-10 in El Paso. The subsections below briefly discuss the observed performance on I-10 EB for the machine learning (ML) model that uses only speed data as an input (rather than speed and volumes) to illustrate our evaluation approach. The results for all models considering MAE, MAPE, and RMSE are provided in separate documents included with this technical memorandum and discussed in the summary section.

10.3.5.4.1. Aggregate Performance

Figure 95 presents the aggregate performance on the analyzed corridor for each considered dataset, and the number of days used to calculate the reported values (sample size). In general, it is desirable for trained models to exhibit similar performance across the testing, validation, and real time datasets, which would suggest that over-fitting is not likely and rule out deployment issues/errors. Figure 95 suggests that, on average, the real-time model is performing as expected. Performance is worse during atypical conditions and on the PM peak period, and in both cases the real-time performance drops with respect to training/testing/validation results. In the case of atypical conditions, the small sample size is likely the reason for the observed discrepancy. The performance deterioration during the PM peak period is interesting and may require further investigation, since it may inform the training process and potentially the model specification.

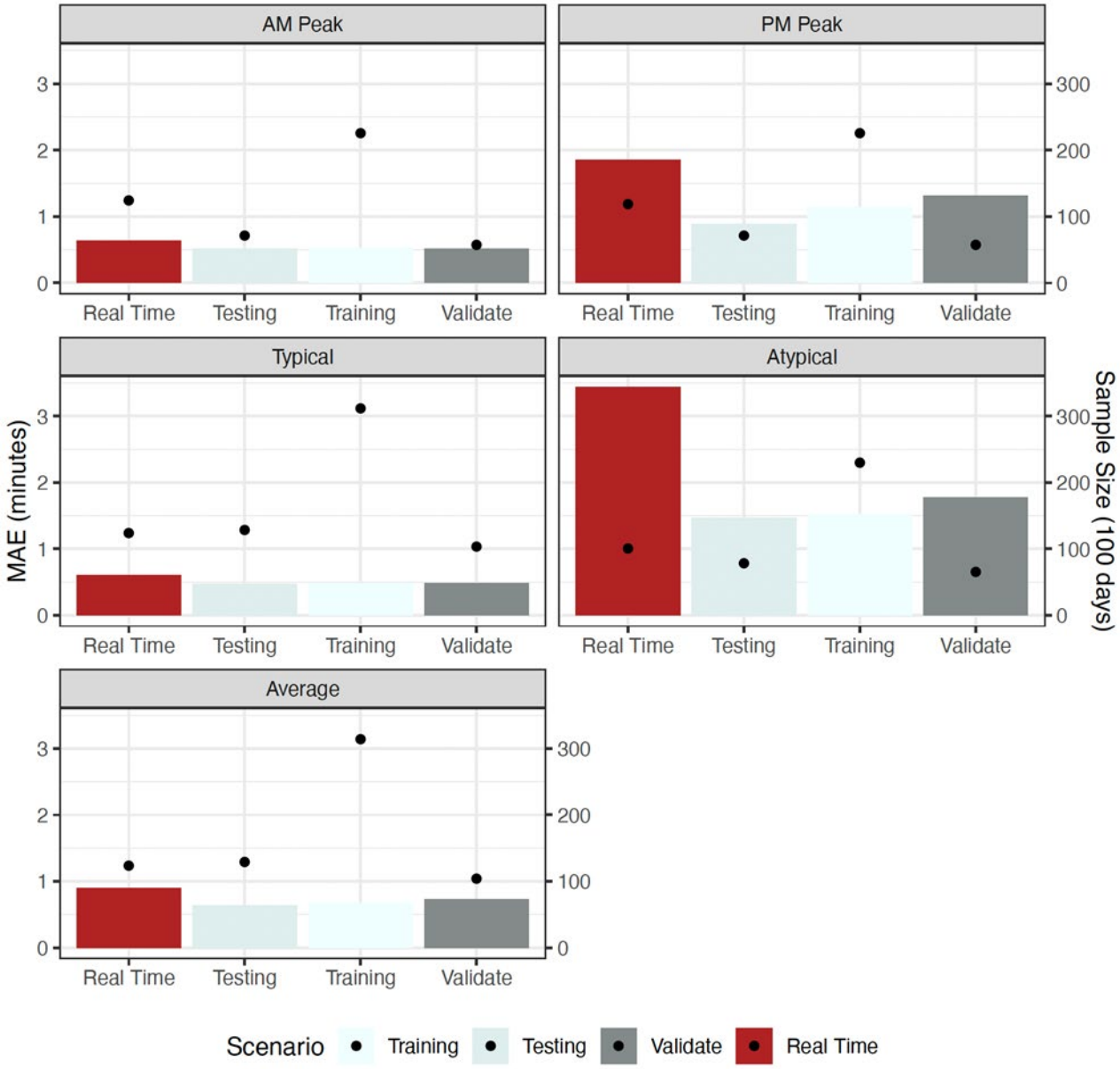


Figure 95: Aggregate performance of selected ML model on I-10 EB. Real-time metrics computed between 11/17/2022 and 03/20/2023. Dots represent the number of data points in each dataset.

10.3.5.4.2. Performance by hour

Figure 96 describes the performance by hour of all models trained on the analyzed corridor and compares it to that of two baseline models: a naïve approach (current practice), which uses observed travel times at time *t* as the estimator of the travel times to be experienced by all travelers departing at time *t*, and "typical" travel times. The latter consists of a static value computed using the training dataset for each 5-minute interval, which represents average conditions. The goal of this research project is to improve upon the performance of baseline models, and this analysis is intended to evaluate the extent to which such goal is met. All

models are clearly better than the use of typical travel times at most hours. ML models are not significantly better than current practice in the observed results, with the exception of some hours during the PM peak period for typical traffic conditions. Errors are fairly small during typical conditions for all models (under two minutes), and very large during atypical conditions.

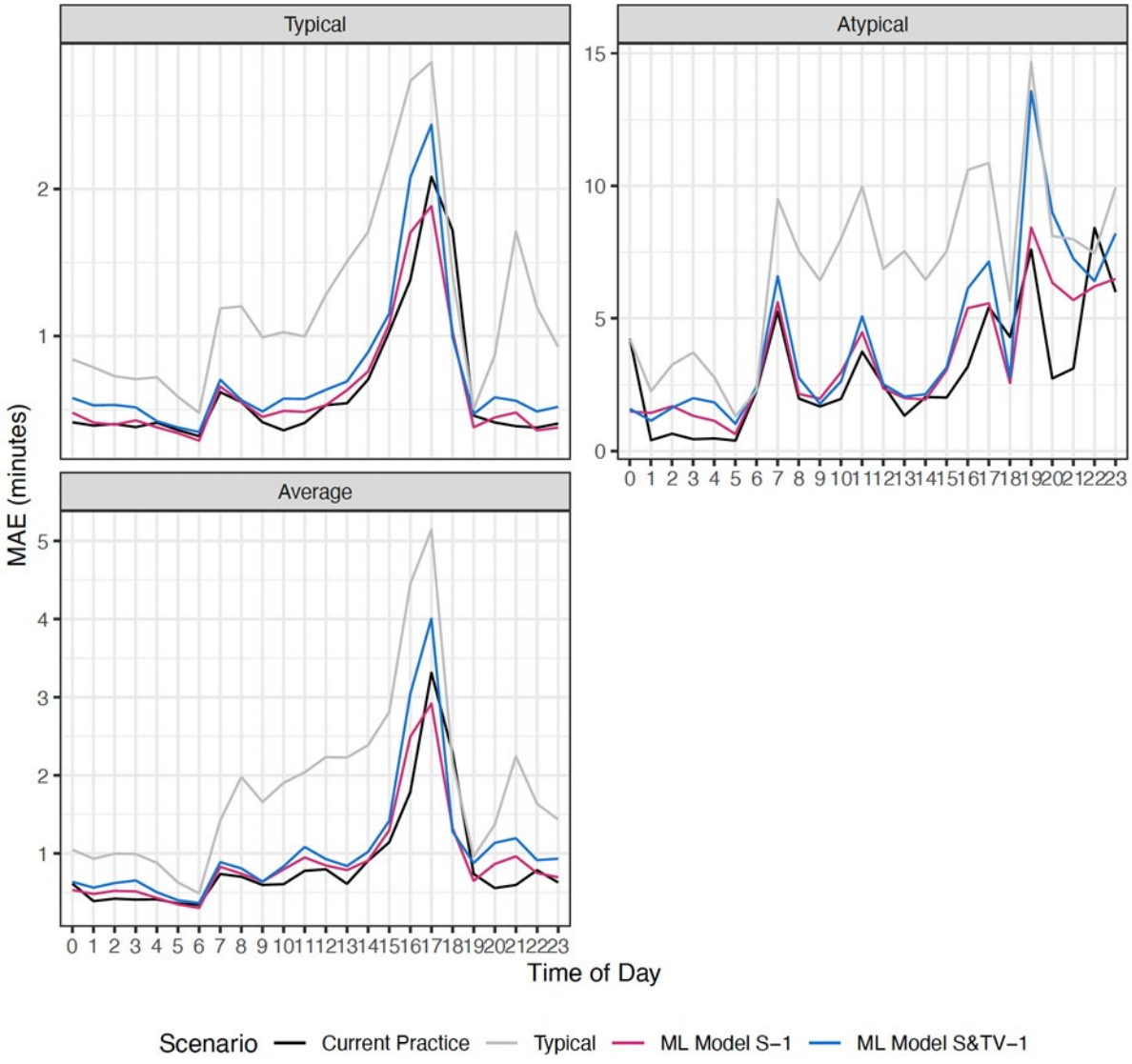


Figure 96: Performance by hour on I-10 EB for real-time models (11/17/2022-03/20/2023).

10.3.5.4.3. Disaggregate performance

Figure 97 illustrates the performance pattern of all models currently running on I-10 EB. Negative error values (in grey) represent an underestimation of experienced travel times, while positive values indicate an overestimation. For all models the pattern suggests under-estimation during congestion build-up in the AM and PM peak periods, and overestimation

when congestion is dissipating. This suggests that the models are not able to anticipate the evolution of traffic quickly enough and may be improved by using a different model specification that breaks down the corridor in smaller sub-sections. Errors in the PM peak period are more pronounced than in the AM peak period for all models, with the ML models being more prone to under-estimation. This visualization also highlights specific dates in which the performance of ML was poorer; it can also identify weekly variations in performance (which are not observed in this corridor but are present in others).

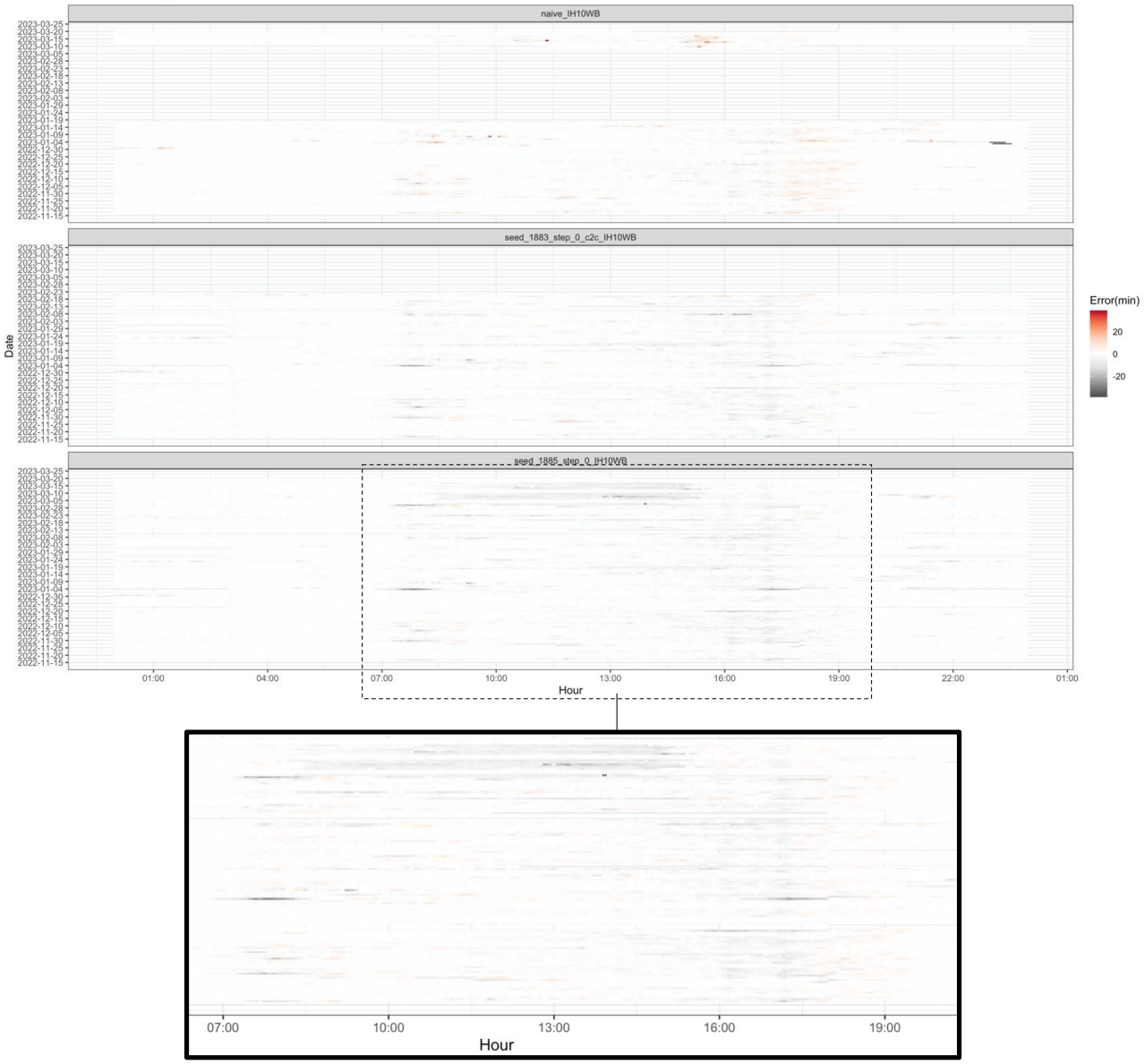


Figure 97: Disaggregate performance (MAE) on I-10 EB 11/17/2022-03/20/2023.

10.3.5.5. Discussion

This section summarizes the approach to be used in the performance evaluation of all initially deployed models and illustrates the computation of mean absolute error (MAE) for one corridor. Results for MAE, MAPE, and RMSE were computed for all models and are provided in separate documents. Our approach is intended to provide a comprehensive understanding of the performance of the model to support model refinement and the development of guidelines for their use. We propose analyses that compare the model performance during the training process to that observed in real time to identify issues such as overfitting and deployment errors. We also compare the performance of ML models to that simpler "baseline" models to better understand the value of the new approach. A visualization of disaggregate data is used to illustrate performance patterns, which is expected to be useful for model refinement and guideline development. This document discusses the results for one corridor in order to illustrate the evaluation approach. The trends observed during our preliminary data analysis consisting of one month of data are consistent with the results observed using four months of data, which suggests that one to two months of data are sufficient to conduct a preliminary assessment of the value of a trained model.

The performance observed up to this point, while promising, does not suggest significant benefits with respect to current practice. In response to the results, CTR identified promising paths to improve performance and trained new models. The data pipelines, database schema, and automated workflows already deployed for this project have been tested and refined through the development of new models. The framework allows for streamlined access to corridor-level metrics and data including travel times and traffic volumes, when available. Besides streamlining the testing and deployment of new ML models, this framework can facilitate the use and interpretation of data by TxDOT for corridors where ML models are not trained or in use, and may be used to visualize and access data using third-party applications similar to the web-based application deployed by CTR.

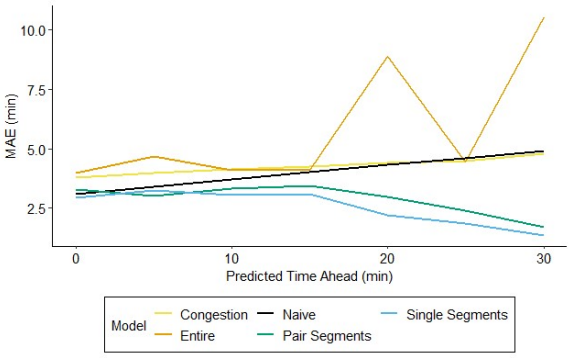
10.3.6. Real-time Evaluation of Phase II models

As with the training results, we present only results for the I-35 SB route. Results from the other routes are provided in Appendix C. We examine the MAE since it is the easiest metric to visualize (figures including the MAPE and an analysis by time of day are provided in the appendix). The plots in Figure 98 show that for this route, the ML models with smaller levels of spatial aggregation are preforming better than the route-level or congestion paths models. The congestion paths model matches the naïve model most closely, suggesting that it is indeed able to capture broad patterns of congestion but less able to predict rapid changes in travel time than the single or pair segments models. The MAE for the best performing ML model (the single segments model) is consistently below three minutes and is also lower at timesteps further into

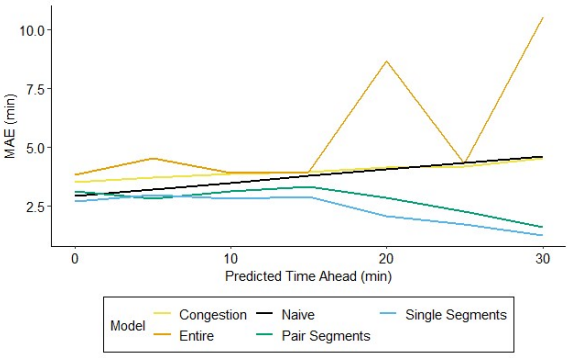
the future. This is an unexpected behavior that needs to be examined more closely. The other three models follow the typical trend of worse performance as predictions progress further into the future.

There are times when all models struggle. During the PM peak period and at times when conditions are atypical, all models tend to have MAE values that are three or more minutes longer than average. The AM peak period also has slightly worse performance, though not nearly as poor as the PM peak. In addition, it is notable that as conditions worsen (become more atypical, congested, or predict further in the future) the spread in the models becomes larger. The worst-case performance is seen by the route-level model during the PM peak with MAE values of almost 20 minutes (for predictions 20 minutes ahead). At that same time, the single segments model has a MAE of only 1.66 minutes, and the naïve model has an MAE of 8.44. This spread shows the importance of choosing the correct model and the potential benefits of the best ML model over the naïve approach.

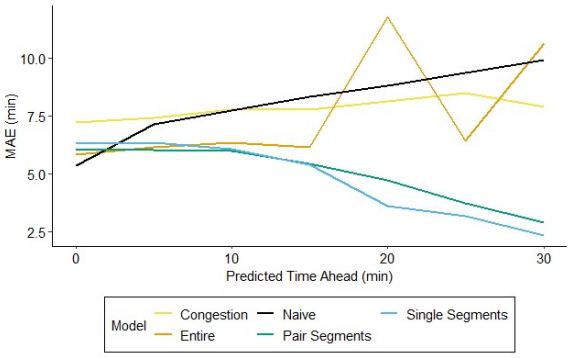
Figure 99 shows the performance of all models by time of day at Step 0 on the selected corridor. While on average the Naïve model is comparable to the segment-level ML model, the performance of the ML model during the morning and peak period is significantly better.



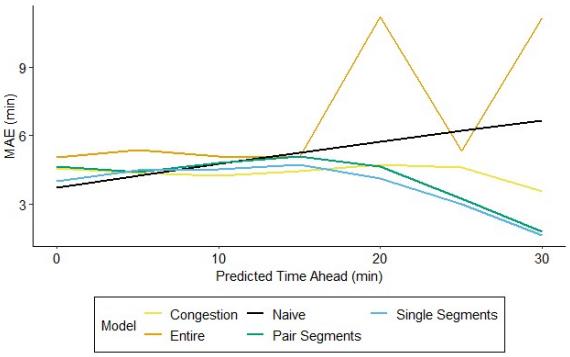
e) I-35 SB average real-time MAE (min).



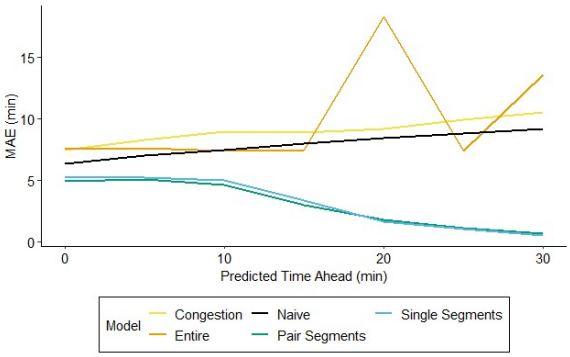
f) I-35 SB real-time MAE (min) for typical conditions.



g) I-35 SB real-time MAE (min) for atypical conditions.



h) I-35 SB real-time MAE (min) for the AM peak period.



i) I-35 SB real-time MAE (min) for the PM peak period.

Figure 98: MAE for naive and ML predictions using real-time data from January 2 - May 31, 2023.

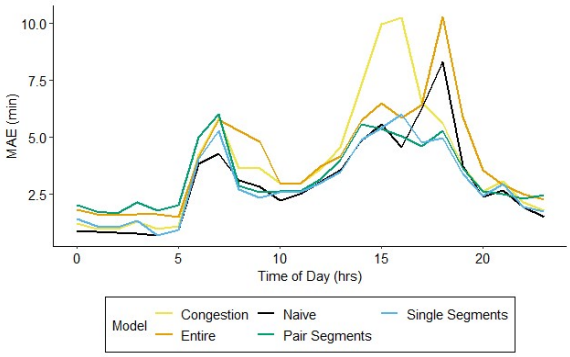


Figure 99. Average Real Time Performance on SB I-35.

10.3.7. Predicting Travel Time Differences Across Competing Routes

For this section, we implement a data-driven approach to quantify the potential benefits of improvements in travel-time predictions in the context of a specific application: providing travel-time information to drivers along alternative routes. Our selected use case is based on the findings of a literature review which suggest that the use of VMS to provide travel times along alternative routes has been observed to provide the most significant benefits (Product 7).

Our evaluation of the value of precise travel-time estimations is based on historical data along the analysis corridors. The historical data is used to compute the travel times that would be predicted using a naïve approach (current practice) and realized travel times, which would correspond to a perfect estimation of experienced travel times. The difference between the naïve model results and realized travel times quantifies the potential benefits of improved travel-time prediction methods. This project developed several ML models to improve upon naïve predictions. Section 9.3.7.2 will demonstrate that models calibrated using the correct level of spatial aggregation can reduce the errors in travel time predictions that we demonstrated for the naïve model.

10.3.7.1. Evaluating the *Potential* Benefits of Accurate Travel Time Predictions in Selected Corridors

Based on review of recent literature, one of the most common and valuable uses of real-time travel-time information provision is assisting drivers choosing between alternative routes. In cases where 2 routes diverge and then converge again there is utility in accurately predicting which route will be faster. The 2 sites examined in this study are the El Paso (ELP) corridors (I-10 and SL 375) and the Austin (AUS) corridors (I-35 and SH 130). Each corridor can have traffic in both directions (EB and WB for El Paso and NB and SB for Austin). See these routes in Figure 76 and Figure 77.

While the overall goal of this project is to use machine learning models to improve existing travel-time prediction practices, this analysis focuses on identifying the maximum possible improvements, which would be attained if travel-time predictions were perfect. Our discussion involves two types of travel times for each corridor, naïve (current practice) and realized travel times (proxy for perfect travel time prediction). The naïve travel-time prediction method involves taking current travel time information on each roadway segment and adding it up along a corridor. The underlying assumption is that segment travel times at the time that a trip starts will remain constant as drivers move through the corridor. The naïve method is typical of models used in practice and predicts that the experienced travel times will be the sum of the current travel times on all segments. We compare this model to the realized travel times, which we compute using historical data and considering the arrival time to each segment along the corridor. We use 5-minute travel times, which implies that drivers arriving at a segment at 9:05 AM may experience a different travel time that those arriving at the same segment at 9:10 AM, which in turn will affect their arrival time to the next segment along the corridor. Comparing naïve and realized travel times allows us to approximate the maximum possible benefits of a machine learning model.

Two approaches are used to analyze the differences between predicted and realized travel times:

- Performance by time of day: This analysis aggregates travel times by time of day allowing us to demonstrate that there are important temporal patterns in travel time prediction that can cause systematic errors when using the naïve approach. This analysis will suggest certain times of the day during which machine learning models can provide the most value. This will also have implications for calibration of the models as these times of day often experience the most erratic changes in travel times.
- Aggregate performance: This approach computes metrics of the average performance over the entire day. Most importantly we report the percentage of the time the realized and naïve travel times would suggest that different routes are faster along the same corridor. This is very important in practice because a VMS reporting these times would misdirect traffic if travel times were not well calibrated, potentially leading to additional congestion.

Results by Time of Day

Each corridor study consists of two routes. We will focus our discussion on the difference between travel times on each route as estimated by the naïve method and the realized travel times. Figure 100 depicts the predicted and realized travel times in the AUS corridor, northbound (NB) direction, on April 1, 2023. There are several notable issues with these predictions. First, around 2 PM there is a major drop and spike in the predicted travel times that is exaggerated in

comparison with realized times. Empirical observations of additional days suggest that this pattern is common, especially as travel times begin to fall after peak periods. The other concern is the period when the two different routes cross; these are times when drivers may be unsure which route to take. A clear example is at 7 PM when the green line for the realized I-35 travel times drops below the SH 130 line but the predicted line does not. In this case drivers would be better off switching from SH 130 to I-35, but the prediction would advise them against this. These periods of inconsistency are critical to minimize and will be the focus of much of the remainder of this TM.

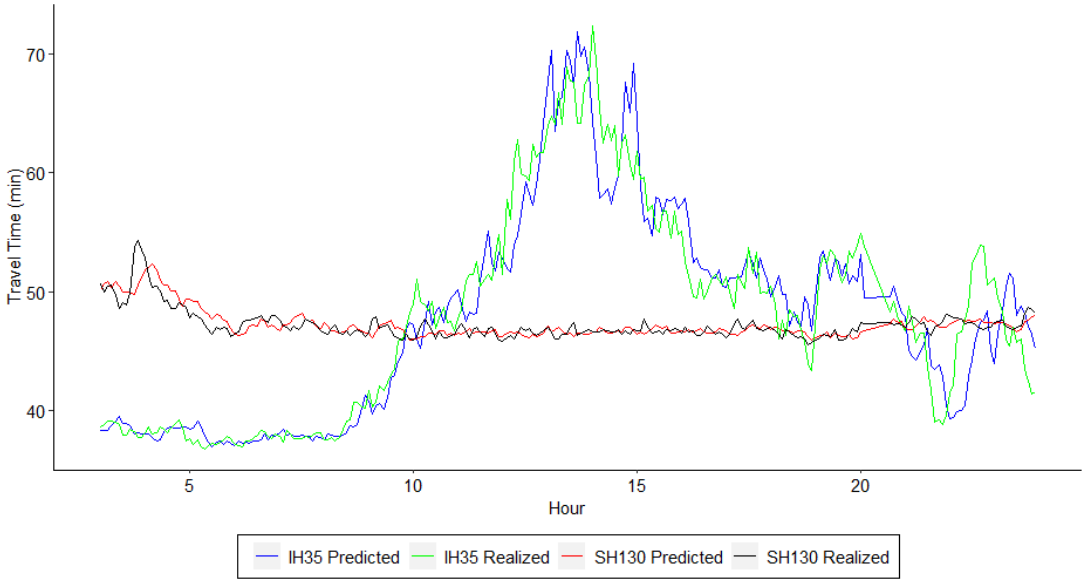


Figure 100: Predicted and realized travel times along the AUS NB corridor on April 1, 2023.

As discussed above, one of the most important reasons why travel-time data is used in practice is to determine the difference between competing routes. Figure 101, Figure 102, and Figure 103 present the difference between travel times predicted using the naïve method (the most used in practice) and realized travel times, as proxy for perfect travel-time predictions. Figure 104 provides a slightly different visualization of the same data that includes corresponding standard deviations. The averages across all days in the study suggest interesting differences and potential areas of improvements. In the El Paso corridors, the differences computed using realized travel times are, for most times of day, more significant than those reported by a naïve prediction method. Providing information that suggests bigger savings in travel times may be more effective when trying to influence driver behavior. Additionally, the predicted difference in travel time is highly variable at night for the El Paso EB corridor, even though the realized travel times are more consistent. In Austin the trend is different but also insightful, with realized travel time suggesting more marked difference during congestion build-up and dissipation periods, where accurate travel-time prediction may be more critical. It is also

interesting to note that at some times of day the direction of the difference in travel times between competing routes is reversed, which suggests that using a naïve prediction method would lead to providing the incorrect information to drivers.

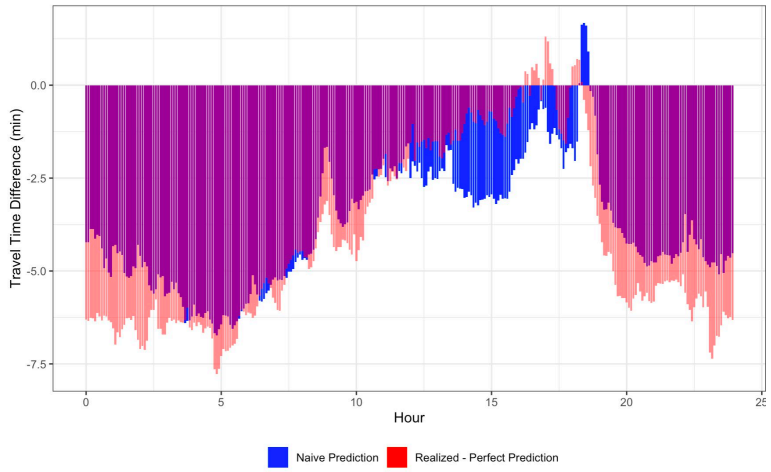


Figure 101: Comparison of travel-time differences between alternative routes computed using a naïve prediction method (current practice) and realized travel time (as a proxy for perfect travel time predictions). El Paso EB (negative means I-10 is faster, positive mean SL 375 is faster).

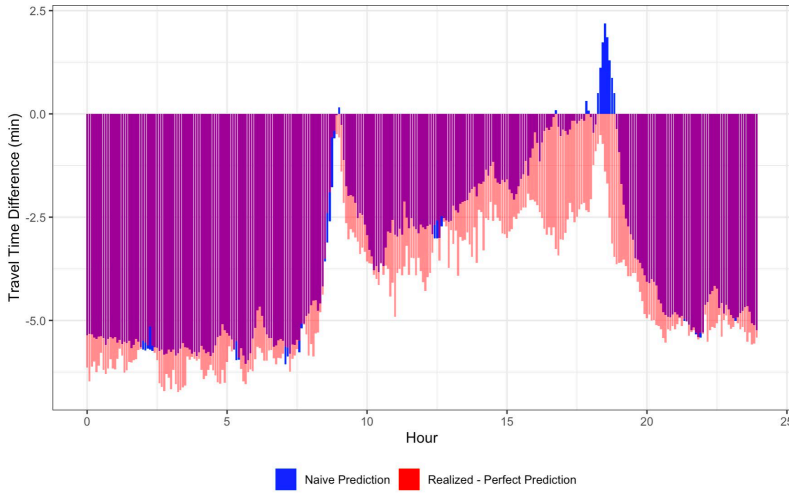


Figure 102: Comparison of travel-time differences between alternative routes computed using a naïve prediction method (current practice) and realized travel time (as a proxy for perfect travel-time predictions). El Paso WB (negative means I-10 is faster, positive means SL 375 is faster).

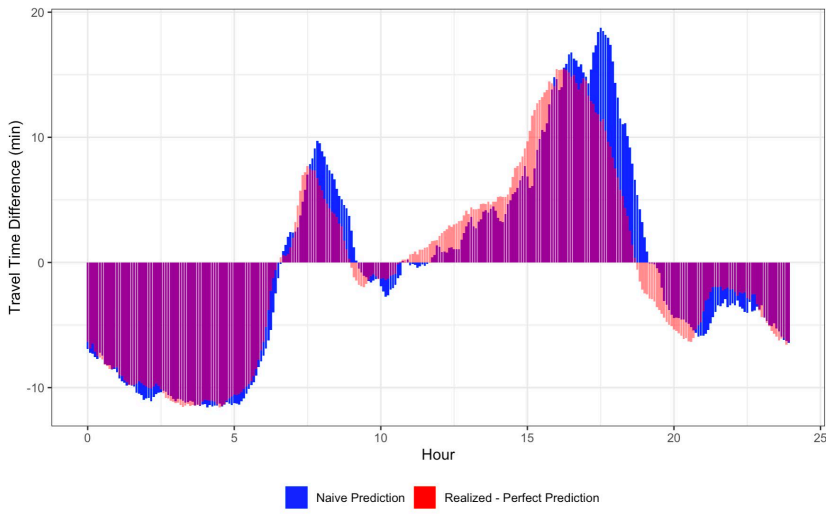
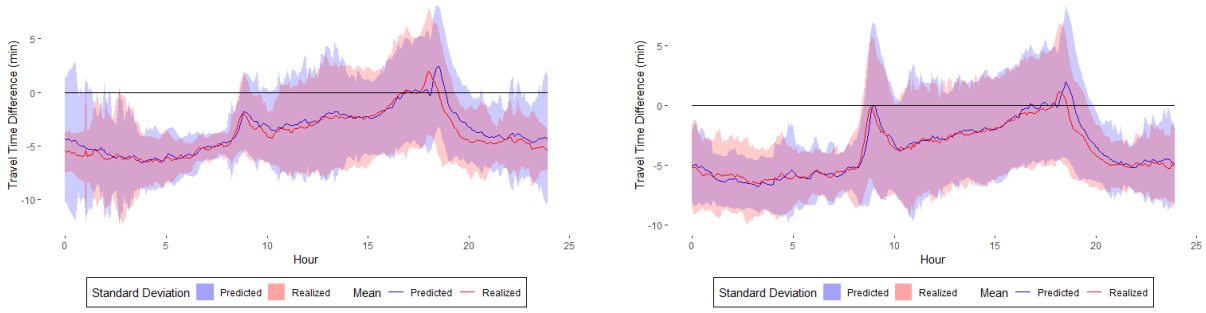
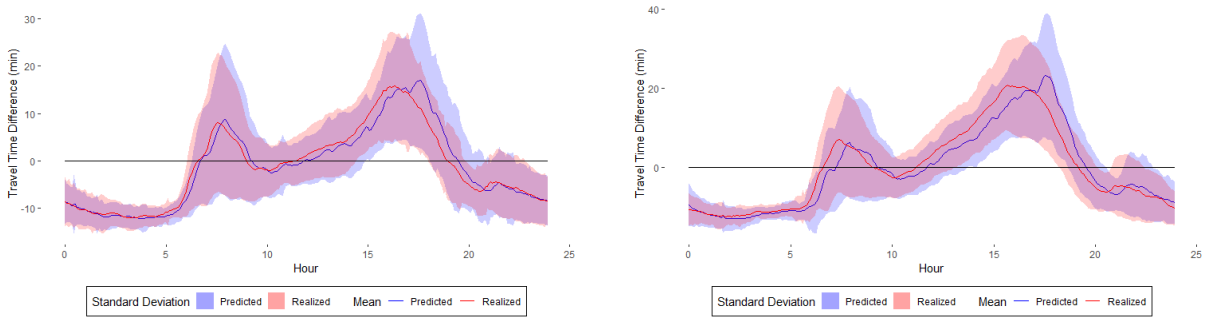


Figure 103: Comparison of travel-time differences between alternative routes computed using a naïve prediction method (current practice) and realized travel time (as a proxy for perfect travel time predictions). Austin NB (negative means I-35 is faster, positive means SH 130 is faster).



a) El Paso EB travel-time difference (negative means I-10 is faster, positive means SL 375 is faster)

b) El Paso WB travel time difference (Negative means I-10 is faster, positive means SL 375 is faster)



c) Austin NB travel-time difference (negative means I-35 is faster, positive means SH 130 is faster)

d) Austin SB travel-time difference (negative means I-35 is faster, positive means SH 130 is faster)

Figure 104: Mean and standard deviation of naïve prediction vs. realized travel-time differences between routes on each corridor.

Though Figure 104 is informative, comparing individual predictions with realized travel times is better. In Figure 105, points are grouped by timestamp and the difference between travel times on the two routes is compared. The plot shows the mean absolute error in the *predicted difference in travel times* between the two routes for all four routes. Note that this is a different metric than the MAE in travel time itself. The plot demonstrates that there are errors in prediction of travel-time difference between routes in excess of seven minutes during the evening peak period. This suggests that as volumes increase (during the morning and evening peaks) the travel times get more difficult to predict. Unfortunately, from a congestion prevention perspective this is the time when it is most important to predict travel times accurately.

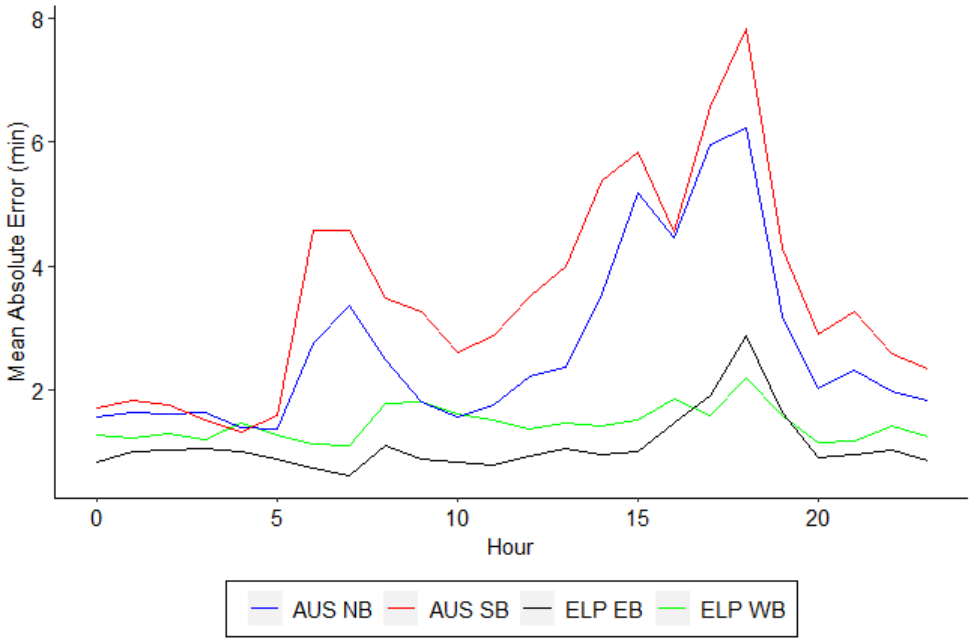
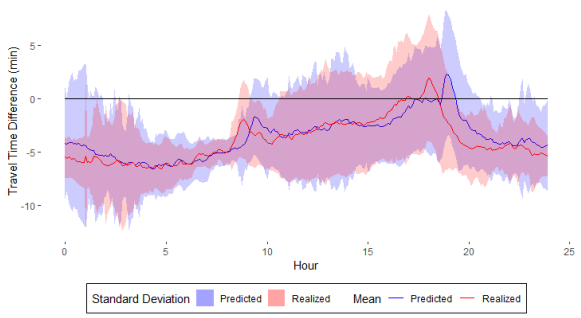


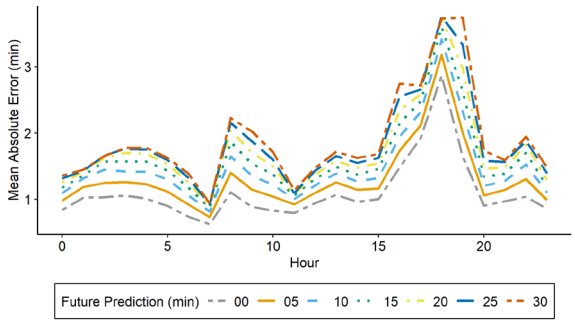
Figure 105: Mean absolute error in naive prediction of difference in travel time between routes.

The results presented above consider a predicted travel time made in real time using the naïve model. Such predictions are generally accurate over short distances and for very short time windows. However, longer term predictions over long corridors are less reliable using the naïve method. We note that there is already an apparent lag between the actual and predicted travel times. This is due to differences at the far end of the corridors. With travel times on the order of 30–60 minutes (and higher during peak periods) there is a large delay between the naïve prediction and the experienced travel time at the end of the corridor. Averaging out these poor predictions with better predictions at the beginning of the corridor leads to the adequate (if slightly delayed) results shown above.

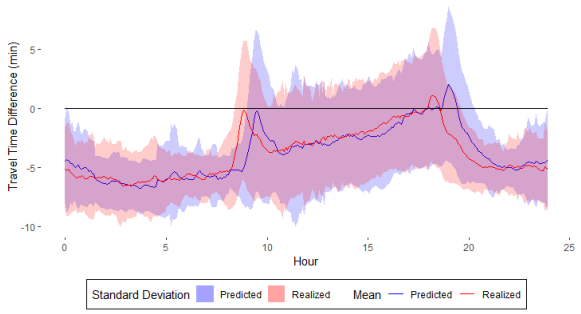
Though there is clearly room for improvement even for real-time predictions, predictions even further in the future are less accurate. Figure 106 depicts the difference in travel time between routes for the naïve model predicted 30 minutes ahead and realized travel times. Average MAE in travel time difference between parallel routes predicted between zero and 30 minutes into the future are also included. Plots a, c, e, and g confirm that there is simply a 30-minute offset from the travel times predicted in Figure 104. In aggregate, the increase in MAE for predictions further into the future demonstrates significant room for improvement if ML models are used. In addition, since these are averages, on any individual day the differences tend to be worse. Section 0 outlines how these empirical observations can be quantified in terms of percentage of times when better predictions will lead to better choices of routes.



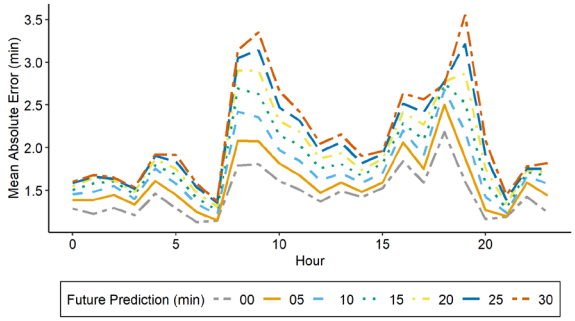
a) El Paso EB travel time difference predicted 30 minutes ahead (Negative means I-10 is faster, positive means SL 375 is faster).



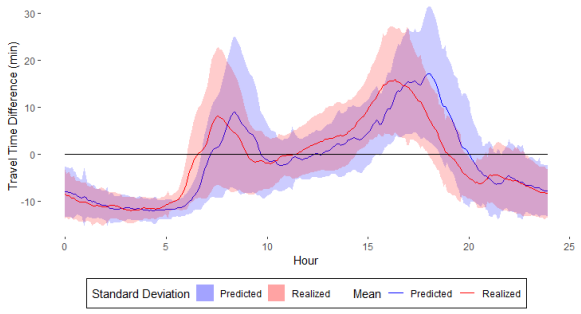
b) El Paso EB mean absolute error in future naive predictions of difference in travel time between routes.



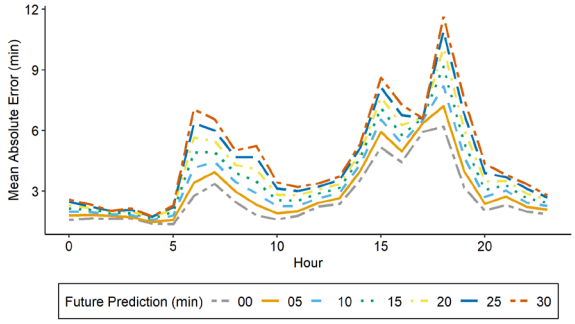
c) El Paso WB travel time difference predicted 30 minutes ahead (negative means I-10 is faster, positive means SL 375 is faster).



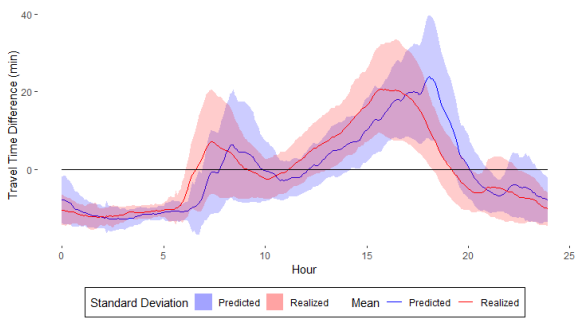
d) El Paso WB mean absolute error in future naive predictions of difference in travel time between routes.



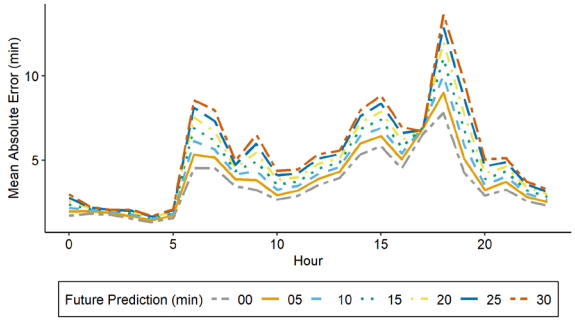
e) Austin NB travel time difference predicted 30 minutes ahead (negative means I-35 is faster, positive means SH 130 is faster).



f) Austin NB mean absolute error in future naive predictions of difference in travel time between routes.



g) Austin SB travel time difference predicted 30 minutes ahead (negative means I-35 is faster, positive means SH 130 is faster).



h) Austin NB mean absolute error in future naive predictions of difference in travel time between routes.

Figure 106: Mean and standard deviation of naïve prediction vs. realized travel-time differences between routes on each corridor with a 30-minute delay (left) and MAE in future naive prediction of difference in travel time between routes at several time increments.

Aggregate Results

Assuming drivers want to take the faster route, the best road to take can be determined each time travel times are predicted by taking the route with the lowest travel time. From the user perspective, the actual travel times predicted are less important than whether they are recommending the correct route. In this section, we determine the best route at each point in time using the naïve and realized travel times. We can then compute the percentage of the time that the naïve model is correctly sending travelers along the correct route.

Table 23 shows the percentage of time the incorrect route was predicted to be the shortest. These percentages are split among times drivers were incorrectly directed to take the primary or secondary routes as defined in Figure 76 and Figure 77.

Taking the Austin NB corridor as an example there were 55,884 total travel-time predictions for the routes along this corridor. Of those, 2,243 incorrectly suggested that I-35 was the faster route while 2,116 incorrectly suggested that SH 130 was faster. Though there are other times when travel-time predictions were incorrect, these cases are the most important because they can lead drivers to make poor decisions about which route to take. Also included are the same values for “atypical” periods when realized travel times were more than 2 standard deviations above or below the mean travel time. As demonstrated in Figure 105, differences between the naïve and realized travel times tend to be highest during peak periods. These periods tend to have a much larger variation in travel time (particularly at the onset and termination of congestion). The majority of atypical conditions happen during those peak periods when supplying accurate information to drivers is most critical.

Table 23: Percentage of time incorrect predictions of shortest route are made using naïve method.

	All Data	Atypical High Travel Times	Atypical Low Travel Times
I-35 NB Predicted Faster	4.01	5.58	7.97
SH 130 NB Predicted Faster	3.79	3.52	5.19
Total AUS NB	**7.80**	**9.10**	**13.16**
I-35 SB Predicted Faster	4.91	5.77	14.12
SH 130SB Predicted Faster	5.00	1.34	10.96
Total AUS SB	**9.91**	**7.11**	**25.08**
I-10 WB Predicted Faster	3.32	8.42	10.00
SL 375 WB Predicted Faster	3.20	3.48	4.12
Total ELP WB	**6.52**	**11.90**	**14.12**
I-10 EB Predicted Faster	2.23	7.77	0.00
SL 375 EB Predicted Faster	2.28	3.74	0.00
Total ELP EB	**4.51**	**11.51**	**0.00**

As above, we can compare the numbers in Table 23 with number for predictions further in the future. Table 24 demonstrates that predictions 30 minutes into the future increase the percentage of incorrect routing guidance by 50–100 percent overall. Even higher increases are seen during periods with atypically high travel times, though fewer incorrect predictions are made in some cases when travel times are atypically low. These cases are dominated by cases with atypically high travel times (323 atypically low data points and 9829 atypically high data points).

Table 24: Percentage of time incorrect predictions 30-minutes in the future of shortest route are made using naïve method.

	All Data	Atypical High Travel Times	Atypical Low Travel Times
I-35 NB Predicted Faster	7.97	11.47	4.00
SH 130 NB Predicted Faster	7.81	4.49	6.40
Total AUS NB	**15.78**	**15.96**	**10.40**
I-35 SB Predicted Faster	8.24	11.46	3.70
SH 130 SB Predicted Faster	6.66	2.07	2.96
Total AUS SB	**14.90**	**13.53**	**6.67**
I-10 WB Predicted Faster	5.50	17.58	8.89
SL 375 WB Predicted Faster	5.40	5.02	11.11
Total ELP WB	**10.90**	**22.60**	**20.00**
I-10 EB Predicted Faster	4.51	21.33	0.00
SL 375 EB Predicted Faster	4.62	5.24	0.00
Total ELP EB	**9.13**	**26.57**	**0.00**

The goal of this section was to acknowledge that an important use of real-time travel-time predictions is to provide information to drivers through active traffic management systems. Based on the literature review, we found that one of the most critical uses of such information is to help drivers decide between competing routes. Even for predictions made in real time using the naïve model we find that the incorrect route is determined to be faster between 4.5 and 9.2 percent of the time. Predictions further into the future increase the errors, and we demonstrate that errors can happen up to 15.79 percent of the time for the worst performing corridor. There is substantial room for ML models to improve these predictions and better direct drivers about optimal routing.

10.3.7.2. Evaluation the *Actual* Benefits of Improved Travel Time Predictions

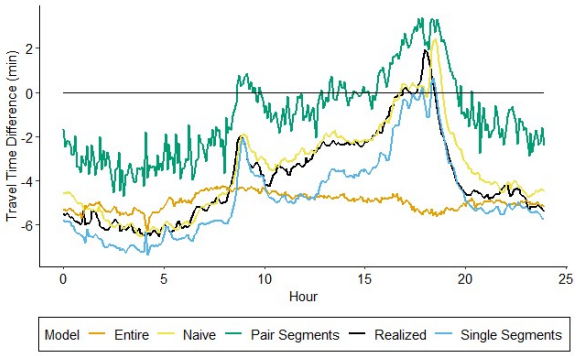
This project developed several ML models to improve upon naïve predictions. This section will demonstrate that models calibrated using the correct level of spatial aggregation can reduce the errors in travel time predictions that we demonstrated for the naïve model.

Results by Time of Day

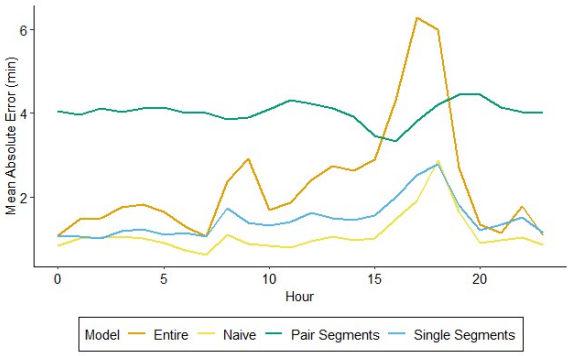
As stated above, the naïve model does very well with predictions in real time or only a short time in the future. Figure 107 shows the same set of plots developed in the previous section, but with the inclusion of the ML models utilizing travel time data only. The plots of the MAE of difference in travel time predictions between competing routes shows that in general the naïve prediction is difficult to beat when predictions are made 0 timesteps ahead. The route-level ML model (which predicts travel times for the entire route) is generally not a good model compared to the other three. Less spatial aggregation tends to be better. Both the single segments and pair segments models perform very close to the naïve model except for the El Paso EB pair segments model. This is expected as each of these models predicts travel times for shorter sections of the corridor and then these predictions are aggregated up to a travel time for the whole route. These models use more computational resources since many ML models must be run and route-level travel-times constructed from the results. However, the process is still sufficiently fast to run in real time. The single and pair segments models better evaluate different patterns of congestion that appear, develop, and propagate separately at different locations.

Figure 108 plots the same figures with predictions 30 minutes ahead. In these figures we can see clear improvements of the ML models over the naïve model. The 30-minute offset in the naïve model causes systematic underprediction of travel times at the onset of congestion and overprediction as congestion dissipates. This trend is shown in the plots of the MAE, where the MAE for the naïve model is generally higher than the pair or single segment models (particularly during and after peak periods). The biggest issue seen in the ML models is shown in Figure 108e

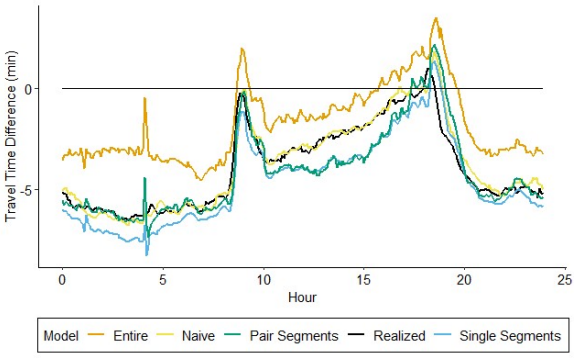
and Figure 108g. Though the MAE in Figure 108f and Figure 108h are low for the pair and single segment models, both models are shown to predict slower travel times on SH 130 relative to the travel times on I-35 than they should. This systematic favoring of one route over another due to incorrect predictions could have important routing implications and should be studied further. Regardless, even during these periods the MAE of these models is still significantly lower than the naïve model.



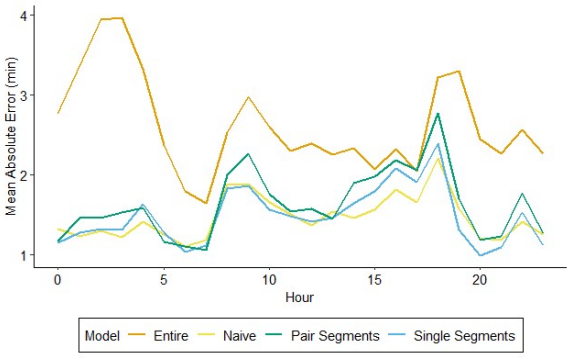
a) El Paso EB ML model travel time difference predicted 0 minutes ahead (negative means I-10 is faster, positive means SL 375 is faster).



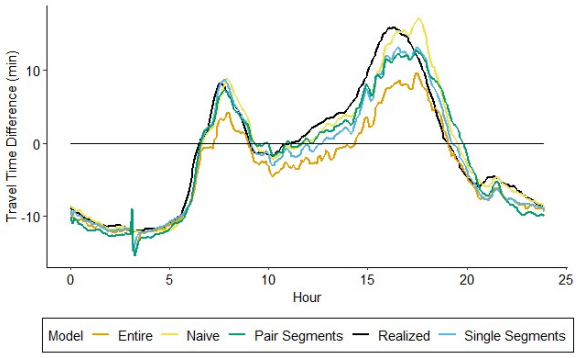
b) El Paso EB mean absolute error in ML model predictions of difference in travel time between routes predicted 0 minutes ahead.



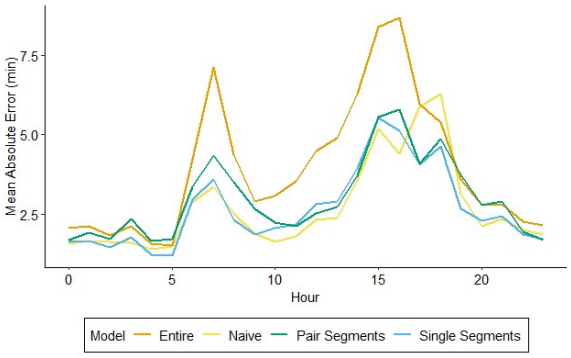
c) El Paso WB ML model travel time difference predicted 0 minutes ahead (negative means I-10 is faster, positive means SL 375 is faster).



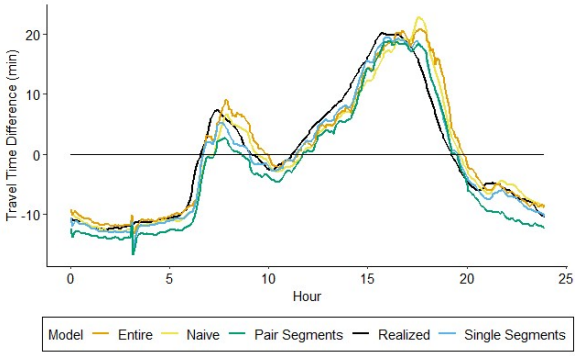
d) El Paso WB mean absolute error in ML model predictions of difference in travel time between routes predicted 0 minutes ahead.



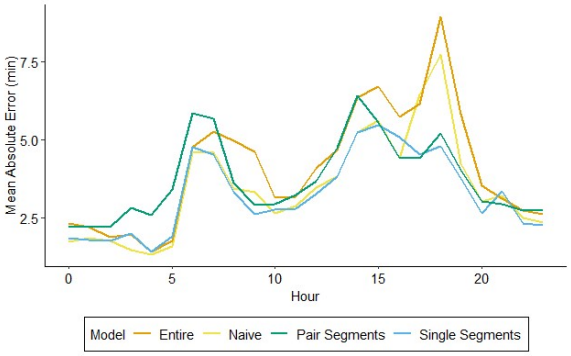
e) Austin NB ML model travel time difference predicted 0 minutes ahead (negative means I-35 is faster, positive means SH 130 is faster).



f) Austin NB mean absolute error in ML model predictions of difference in travel time between routes predicted 0 minutes ahead.

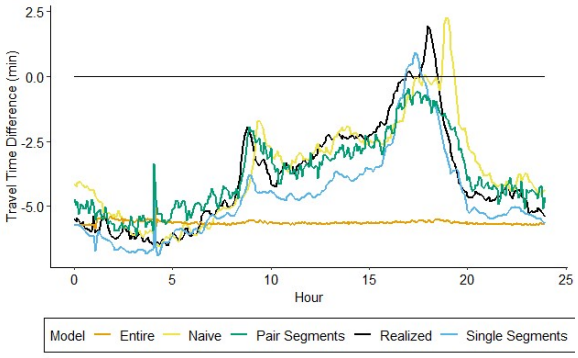


g) Austin SB ML model travel time difference predicted 0 minutes ahead (negative means I-35 is faster, positive means SH 130 is faster).

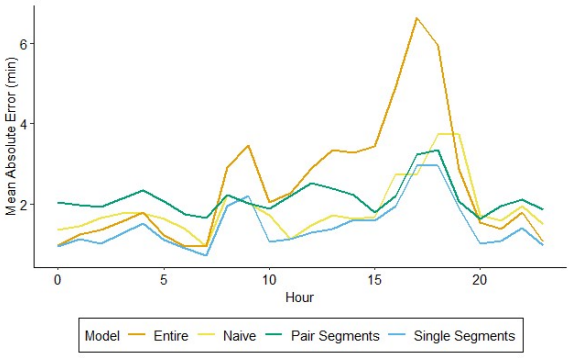


h) Austin NB mean absolute error in ML model predictions of difference in travel time between routes predicted 0 minutes ahead.

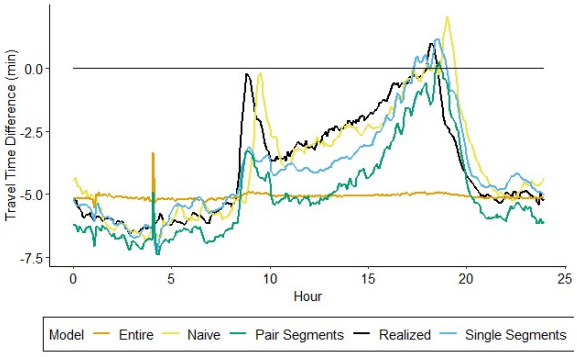
Figure 107: Mean of ML model prediction vs realized travel time differences between routes on each corridor with a zero-minute delay (left) and MAE in ML model prediction of difference in travel time between routes.



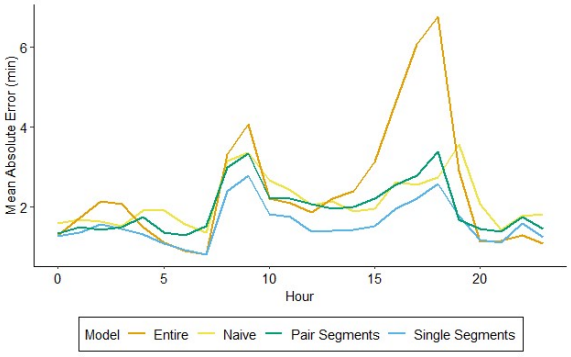
a) El Paso EB ML model travel time difference predicted 30 minutes ahead (negative means I-10 is faster, positive means SL 375 is faster).



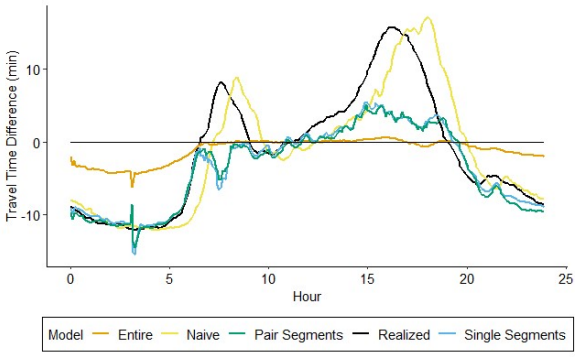
b) El Paso EB mean absolute error in ML model predictions of difference in travel time between routes predicted 30 minutes ahead.



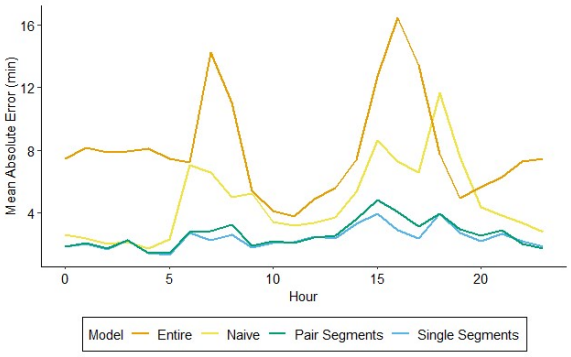
c) El Paso WB ML model travel time difference predicted 30 minutes ahead (negative means I-10 is faster, positive means SL375 is faster).



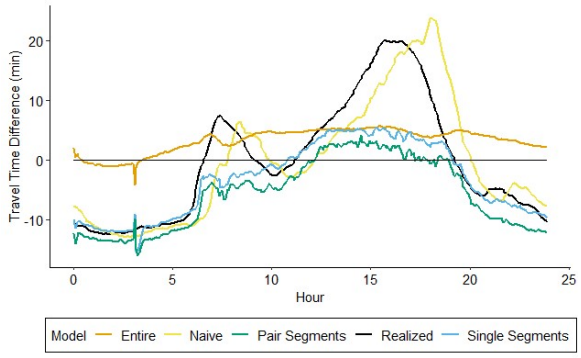
d) El Paso WB mean absolute error in ML model predictions of difference in travel time between routes predicted 30 minutes ahead.



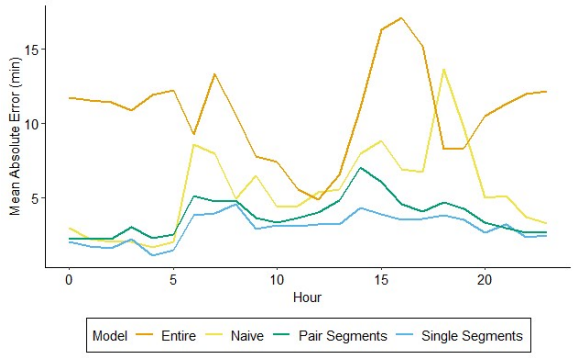
e) Austin NB ML model travel time difference predicted 30 minutes ahead (negative means I-35 is faster, positive means SH 130 is faster).



f) Austin NB mean absolute error in ML model predictions of difference in travel time between routes predicted 30 minutes ahead.



g) *AUS SB ML model travel time difference predicted 30 minutes ahead (Negative means I-35 is faster, positive means SH 130 is faster).*



h) *AUS NB mean absolute error in ML model predictions of difference in travel time between routes predicted 30 minutes ahead.*

Figure 108: Mean of ML model prediction vs realized travel time differences between routes on each corridor with a 30-minute delay (left) and MAE in ML model prediction of difference in travel time between routes.

Aggregate Results

Finally, we discuss the same aggregate model of performance previously developed (the percentage of time travel times predictions indicate the incorrect route to be faster). Overall, these results confirm the findings above that the ML models represent an improvement over the naïve model. From Table 25, first note that these decisions are more difficult to predict in the Austin area (due to the high levels of congestion on I-35 during peak periods). As expected, predictions made more timesteps ahead are consistently less accurate. Unfortunately, the route-level model continues to perform poorly using the metric for the same reasons as above. The aggregate travel times across entire corridors incorporate more complicated interactions between different congestion propagation mechanisms.

The single and pair segments models perform better, though the pair segments model fails for the El Paso EB corridor even with predictions closer in time. This is surprising as these predictions are generally easier. However, the single segments model performs better than the naïve model in almost every case. In every corridor the percentage of incorrect predictions is less than 10 percent for the single segments model, while the naïve model reaches an almost 15 percent incorrect prediction rate for the Austin SB corridor 30 minutes ahead. Additional tables for atypical conditions are included in the appendix.

Table 25: Comparison of percentage of time incorrect predictions of shortest route are made using naive method versus ML models.

Model	Timestep	All Data											
		EB			WB			NB			SB		
		I-10	SL 375	**ELP**	I-10	SL 375	**ELP**	I-35	SH 130	**AUS**	I-35	SH 130	**AUS**
Naïve	0	2.23	2.28	**4.51**	3.32	3.20	**6.52**	4.01	3.79	**7.80**	11.23	5.00	**9.17**
	5	2.70	2.76	**5.46**	3.46	3.34	**6.81**	4.69	4.46	**9.16**	11.82	5.02	**10.24**
	10	3.10	3.17	**6.27**	3.94	3.81	**7.75**	5.41	5.20	**10.61**	12.11	5.30	**11.45**
	15	3.51	3.59	**7.10**	4.37	4.25	**8.62**	6.10	5.91	**12.02**	12.41	5.60	**12.62**
	20	3.90	3.98	**7.88**	4.76	4.65	**9.42**	6.75	6.58	**13.33**	12.75	5.94	**13.72**
	25	4.21	4.30	**8.51**	5.13	5.02	**10.15**	7.38	7.21	**14.59**	13.10	6.29	**14.76**
	30	4.52	4.62	**9.14**	5.50	5.40	**10.90**	7.97	7.81	**15.78**	13.48	6.66	**15.78**
Route-Level	0	10.97	0.03	**11.00**	1.57	8.71	**10.28**	12.20	1.80	**13.99**	19.54	1.99	**9.53**
	5	0.20	64.29	**64.49**	1.26	12.49	**13.75**	4.62	6.10	**10.73**	9.90	5.95	**10.31**
	10	0.42	63.64	**64.06**	12.93	0.00	**12.93**	8.20	3.16	**11.36**	15.37	3.23	**10.62**
	15	11.38	0.00	**11.38**	4.12	5.10	**9.22**	7.38	5.50	**12.89**	13.34	4.95	**12.41**
	20	11.38	0.00	**11.38**	12.93	0.00	**12.93**	10.50	3.93	**14.43**	16.68	3.67	**40.94**
	25	11.38	0.00	**11.38**	12.93	0.00	**12.93**	10.04	3.36	**13.40**	17.29	3.02	**10.73**
	30	11.38	0.00	**11.38**	12.93	0.00	**12.93**	13.57	9.44	**23.01**	16.12	11.24	**50.06**
Single Segments	0	3.44	2.44	**5.88**	5.16	1.68	**6.85**	4.04	3.15	**7.19**	11.84	3.68	**7.04**
	5	3.24	2.71	**5.96**	4.09	1.73	**5.82**	5.20	2.27	**7.47**	13.72	2.76	**7.19**
	10	3.29	2.36	**5.65**	2.41	3.57	**5.98**	6.72	1.79	**8.51**	14.30	2.55	**7.78**
	15	2.72	3.45	**6.18**	3.27	2.91	**6.19**	4.59	2.79	**7.39**	11.96	3.32	**8.12**
	20	3.59	2.39	**5.98**	4.47	1.96	**6.44**	4.43	2.74	**7.17**	11.86	3.29	**8.28**
	25	5.37	1.32	**6.69**	4.27	2.58	**6.85**	4.34	3.45	**7.79**	11.82	3.80	**9.11**
	30	4.94	1.68	**6.63**	5.47	1.92	**7.39**	3.53	5.46	**8.98**	11.21	5.06	**10.04**
Pair Segments	0	1.50	39.80	**41.31**	3.85	3.26	**7.11**	3.84	3.92	**7.76**	9.61	4.44	**7.59**
	5	3.16	13.99	**17.15**	2.89	3.97	**6.85**	2.47	5.86	**8.33**	8.43	4.85	**9.34**
	10	2.41	23.30	**25.71**	4.01	3.54	**7.55**	4.51	3.44	**7.95**	11.89	3.10	**11.37**
	15	3.86	6.86	**10.73**	4.85	2.25	**7.10**	3.51	3.64	**7.15**	10.37	3.57	**9.69**
	20	6.20	3.27	**9.47**	5.25	3.14	**8.39**	2.69	4.96	**7.64**	9.09	4.14	**9.96**
	25	7.33	0.98	**8.31**	6.00	2.32	**8.31**	4.03	3.57	**7.60**	10.06	3.78	**10.75**
	30	5.41	2.10	**7.51**	8.03	1.43	**9.45**	4.10	4.73	**8.83**	10.57	4.30	**10.79**

10.3.7.3. Summary

This section provided a literature review on the use of VMS to communicate real-time information to drivers, which suggests that the provision of travel times along competing routes is among the most valuable applications. Though findings suggest that the precision of travel-time information is not necessarily critical for all applications, it is important for it to be accurate when drivers are choosing between alternate routes. We use this as the primary

metric for assessing the value of high-precision travel-time estimates, which we expect can be provided using machine learning models. Our approach finds the maximum possible improvements by comparing naïve travel time predictions, commonly used in practice, to realized travel times, which would correspond to a perfect estimation of travel times. Results from1 year of historical data suggest that there are times of day (particularly peak periods and atypical traffic conditions) during which drivers may be directed onto incorrect routes using the naïve prediction method. Specifically, over 8 percent of all predictions using the naïve method directed drivers onto incorrect routes on the Austin NB corridor. This is a large number of drivers taking sub-optimal routes and suggests that better machine learning models are needed to more accurately predict travel times during windows of uncertainty.

10.4. Conclusions

This chapter describes the development and performance of three sets of models, including performance on the training dataset and when deployed in real time.

- Preliminary models were developed as a proof of concept on I-35 using one year of probe-based speed data and traffic volumes available form smart work zone trailers (which are no longer available). A recurrent neural network (RNN) model was trained for each INRIX segment in order to predict travel times up to one hour into the future at five-minute steps. Total corridor travel time was computed dynamically, by considering arrival time at each successive segment along the route. Results were very promising, with errors almost 40 percent lower than those that would result from using a naïve approach that simply adds the travel time along all segments at the time that the trip starts.

- Models at four additional sites were trained for field testing, which involved developing a framework to archive and efficiently access training data, real-time data, model predictions and realized travel times. Limited volume data was available at the selected sites and the results obtained when using it did not show improvement upon models that used only probe-based data (Phase I models). Further research may consider additional analysis of the quality and aggregation of the volume data. In Phase II, researchers explore different levels of spatial aggregation for model training, seeking to reduce the associated computational effort while improving model fit. Models were trained at the route-level, sub-route level, segment level and pair-of-segments level.

    - Smaller spatial aggregations led to better results in general, which in some cases outperformed the naïve approach. Improvements upon current practice were not as significant as those observed in the preliminary models, which suggest that the incorporation of volume data may be critical to maximize the value of

these models. Location-specific traffic patterns may also play a role in the potential value of enhanced travel time prediction methods, which may be more valuable on longer corridors where traffic conditions may change rapidly.

- Models developed to examine differences in spatial aggregation demonstrate benefits of low levels of aggregation. Though the R^2 values of training data are lower for predictions at the lowest time (0.57 for the single segments model versus 0.88 for the entire route) on I-35 SB, when predicting farther in the future the accuracy of predictions is much more stable (0.33 versus 0.04 for the same models).
- Machine learning models also showed benefits when used to provide comparative travel times across routes. In every corridor the percentage of incorrect predictions is less than 10 percent for the single segments model, while the naïve model reaches an almost 15 percent incorrect prediction rate for the Austin SB corridor 30 minutes ahead.

10.5. Pathway to Implementation

The naïve model had a MAE in difference in predicted route travel times ranging from 1.11 minutes on the ELP EB corridor to 3.58 minutes on the AUS NB corridor. For the best performing ML model (single segments), the range was smaller, but the values were comparable: 1.47-3.25 minutes on the same corridors. When predicting 30 minutes into the future, the naïve model is outperformed substantially by the ML models 1.98-5.63 minutes versus 1.47-2.80 minutes respectively. As part of project 0-7034, the research team worked on the development of models for short-term travel time prediction (STTTP) on freeway corridors. STTTP involves forecasting travel times one or two hours into the future at a fine-grained temporal resolution (five to 15 minutes). Predictions are often updated every few minutes and are necessary to estimate realistic end-to-end travel times along freeway sections, particularly when some segments in the analyzed section experience congestion. The methods used in this project have been trained and tested on a section of I-35 through Austin using INRIX speeds and volume data collected by smart work zone trailers. Preliminary results based on one year of data from 2019 are very promising, with machine learning models outperforming traditional approaches by more than 40 percent during the AM and PM peak periods.

The ability to accurately estimate experienced travel times in near-real time can support enhanced traffic management solutions for reducing congestion and user-delay cost. If the models are used to provide information through direct messaging signs (DMS), travelers may be diverted to less congested corridors more effectively, potentially improving travel times and safety. Traffic operators may also consider using real-time travel time predictions to adjust

signal timing plans during atypical traffic conditions, which may further improve speeds and safety.

Therefore, further real-world environment testing of the machine-learning-based methods for STTTP developed in this effort is highly desirable. Successful implementation will include the following components:

- Refining use cases for improved short-term travel time prediction, such as public information dissemination and traffic operations support.
- Developing sustainable workflows for model training at new sites.
- Implementing the necessary pipelines to generate predictions in real-time.
- Developing a continuous evaluation framework to identify conditions under which models should not be used and/or suggest model re-training based on changing conditions.
- Evaluating the model performance over a prolonged period.

Specific tasks are summarized in Figure 109 and will be discussed in the reminder of this document.

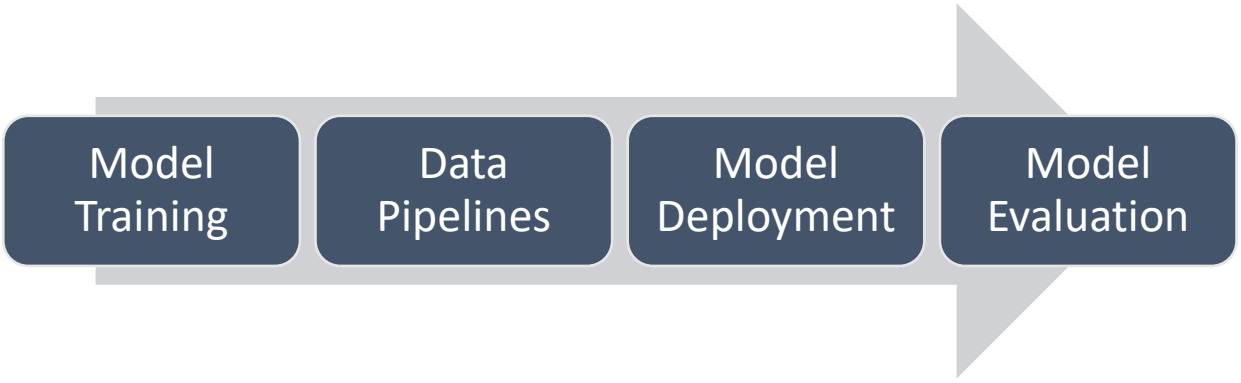


Figure 109. Implementation Tasks.

10.5.1. Model Training

For field implementation, the models developed in this effort should be trained following the methods already tested for project 0-7034 and using data corresponding to the selected implementation site. The training process considers two types of data: probe-based speed data, as provided by INRIX, and traffic volume data from several points throughout the analyzed corridor, which in this effort was obtained from smart work zone trailers. The training process involves collecting and processing historical data, running the training scripts to identify model parameters, and reviewing the final model performance to assess the need to further refine the

selection of hyper-parameters. The following section describes important considerations to develop a sustainable and replicable training process.

10.5.1.1. Temporal and spatial resolution of data

The temporal and spatial aggregation of training and deployment data is likely to have an impact on model performance. The data used to train the original models was available at one-minute time intervals and was further aggregated for model training and testing. Probe based speed data is expected to be available for most of the facilities managed by TxDOT at the desired aggregation level. Automatically collected traffic volume data is likely to be available at a similar temporal resolution. We recommend aggregating data into five- to 15-minute intervals for model training and deployment, and if possible further exploring the impact of aggregation on model performance.

The spatial aggregation of probe-based data is defined by the length of the segments (referred to as XD segments) over which INRIX speeds and travel times are reported. XD segments in rural areas may be longer than those used for original model training, and the impact of such difference on model performance requires further analysis. Continuously collected vehicular volume data is not as widely available as probe-based speed data. The models developed for project 0-7034 leveraged data from sensors that were placed one to two miles apart and covered most of the analyzed corridor. The availability of volume data at new sites may impact model performance. Models can be trained using speed data alone but are expected to perform better when using volume and speed data.

10.5.2. Collection of historical data

Probe-based speed data vendors often provide seamless access to historical data. The models developed for this project were trained and tested using one year of probe-based data, which is expected to capture a range of traffic conditions, along with daily and seasonal patterns. At the time this project was competed, researchers also had access to one year of traffic volume data, which was archived from a separate project. Traffic volume data collected by temporary or permanent ITS sensors is not necessarily archived, and historical data may not be available at new deployment sites. Researchers have started a conversation with TxDOT's ITD in order to understand the prospective availability of historical ITS data, but near-term deployments may need to consider setting up a custom pipeline to archive ITS data at selected deployment sites. Models may be initially trained using only probe-based speed data and refined after sufficient volume data has been collected.

10.5.3. Automation and refinement of model training

The scripts used to train the models developed in this project will be provided as a deliverable, along with documentation describing how the data should be pre-processed to be used with the scripts. Data pre-processing can be automated to support a fully automated training process. Model training involves identifying a set of parameters to be used with real-time data in order to produce accurate travel time forecasts.

It is important to consider that the methods used for model training are controlled by a number of variables, or "hyperparameters." The selection of hyperparameters plays a critical role in model performance. The training process proposed in this effort identifies hyperparameters by systematically testing model performance for a range of potential hyperparameter values. If model performance after training is not satisfactory, it may be necessary to adjust the hyperparameter selection process, which involves refining the scripts developed during 0-7034.

10.5.4. Data Pipelines

Model deployment will require access to archived data for training and to real-time data to generate travel time predictions. Predictions will be generated on a continuous basis and may be streamed in different ways depending on the specific implementation. Figure 110 provides a schematic description of the training and deployment data workflows.

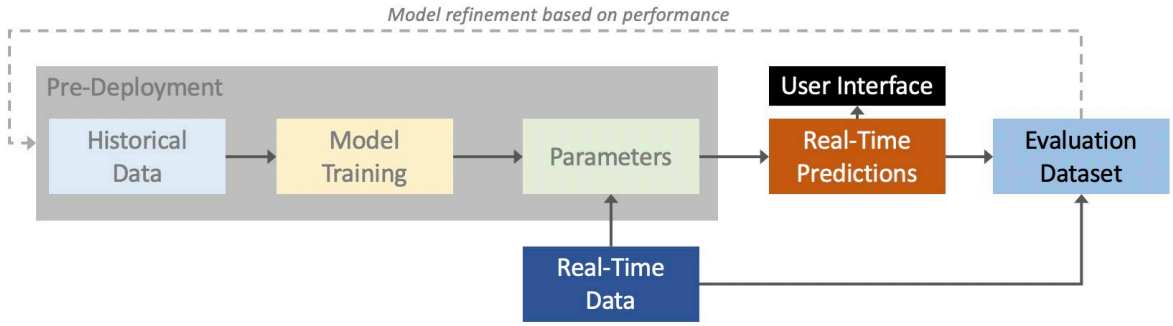


Figure 110. Data Workflows.

The use of historical data for model training was described in the previous section and may be accomplished using a data sample consisting of one or more years of data provided by the same sources to be used after the models are deployed. Although a sample is sufficient for initial model training, the data pipeline suggested below also archives data which may be used to refine models after deployment.

Probe-based speed data is often accessed in real time using the application programming interface (API) available from the vendor. The pipeline development requires identifying the segments for which data is needed and developing the scripts to retrieve and archive it.

Real-time access to volume data is expected to be site-dependent, with the center-to-center (C2C) feed maintained by TxDOT being a potential access point for data collected by permanent ITS sensors. Data collected by temporary sensors, such as smart work zone trailers, may also be streamed through the C2C feed if vendors are willing/able to use the protocol. The process of retrieving and archiving this data may be automated once access to the appropriate data stream is obtained.

Travel time predictions will be generated in real time using speed and, potentially, volume data and they may be streamed via API or through a web interface for their use by relevant TxDOT staff. It is also desirable to archive predictions during a trial period for model evaluation and report generation.

10.5.5. Model Deployment

Model deployment involves using the data pipelines described in the previous section to produce travel time forecasts, which may be presented to travelers and/or decision makers. It is critical to clearly define the use case for the produced travel time forecasts in order to design an effective approach to deliver the information and a meaningful evaluation process.

After the model predictions are made available in real time, a testing and evaluation process should be used to understand:

- Average prediction accuracy and need for model refinements
- Scenarios under which predictions should not be used
- Benefits derived from improved predictions

10.5.6. Model Evaluation

Researchers should define and compute metrics to evaluate the performance of the proposed algorithms over a three to six month period, considering factors such as varying traffic and weather conditions and the presence of data issues. In addition to evaluating prediction accuracy, the evaluation framework may include a component to assess the value of improved forecasts. Comparisons with predictions produced by simpler methods and/or other sources of similar data may be desirable in this context, along with other metrics specific to the selected use case.

In addition to assessing average model performance during a trial period, it is desirable to develop a workflow to continuously evaluate model performance in order to identify scenarios in which the models should not be used, as well as the need to re-train models to improve overall performance.

10.5.7. Summary and Discussion of Required Computational Resources

This document provides a high-level description of the steps and considerations needed to deploy the short-term travel time prediction models (STTTP) developed for project 0-7034 in a real-world environment.

The development of adequate pipelines to retrieve and archive probe-based speed data and traffic volumes streamed in real time is a critical component of the implementation process. The effort involved in such a step will depend on the access to traffic volume data at the selected implementation site, since probe-based speed data is expected to be readily available from INRIX at most facilities managed by TxDOT. Model training requires access to historical data, which is also expected to be straightforward in the case of probe-based speed data but may be more challenging for traffic volume data. Users may need to consider training and deploying models using only speed data and begin the collection of historical volume data during the trial period. Models may be re-trained and refined based on performance after sufficient volume data has been collected.

Model deployment involves providing access to travel time forecasts in real time; the approach used to provide such access will be dependent on the specific use case (e.g., public information dissemination or real-time traffic management). For model evaluation, we suggest a comprehensive approach that includes a continuous evaluation framework, as well as the definition of metrics that consider not only the accuracy of the travel time forecasts, but the value of more accurate forecasts in the context of the selected use case.

Implementing the research conducted for project 0-7034 also requires adequate computational resources to support model training, data archival and analysis, and the generation of travel time predictions.

- Model training will require the most computational resources and it will need to be performed once at the beginning of the project and may be repeated again to address observed issues, incorporate new data sources, or ensure optimal performance. Training will be most efficient if it is performed on a system that can leverage graphic-processing-units (GPUs).

- While real time travel time predictions involve minimal computational effort, providing access to forecasts will require developing an interface suitable to the selected use case. Web-based interfaces, which will require access to a hosting service, are a convenient way to provide data access to decision makers.
- Data archiving is recommended, including that of both real-time data and travel time predictions. If storage space is a concern, travel time predictions may be re-created for testing and evaluation purposes using archived input data. Historical probe-based speed data is typically accessible on demand, which makes traffic volume data the only data type that may require dedicated archival workflows. Relational databases may be used to facilitate data analysis for all data types. A Postgres server was used for project 0-7034.

Cloud-based services may be used to support all the necessary steps for model deployment. Some of the source code and scripts developed and documented in this effort may be used and refined for implementation. Data pipelines developed to manage INRIX data or data broadcasted through the C2C feed are likely to be straightforward and easy to replicate if properly documented during implementation, which is expected to facilitate further model testing and refinement.

The implementation of the models developed in project 0-7034 can improve TxDOT's ability to accurately estimate experienced travel times in near-real time, which may support enhanced traffic management solutions, mitigate congestion, and lead to significant user-delay cost reduction.

Chapter 11. Conclusions

This project explored the use and value of artificial intelligence and machine learning (ML) in transportation taking a multi-pronged approach that includes a literature review, a workshop, a survey, the development of three prototype ML models for four high priority use cases, and the field testing of one of the prototyped models. Initial tasks were exploratory in nature and led to a better understanding of the current and prospective uses of ML in transportation, corresponding data needs, and specific use cases of interest to TxDOT.

The literature review provided a background on artificial intelligence techniques and tools, and explored transportation applications in system and service planning, operations, asset management, public safety and enforcement, communications, and business administration. Researchers found that there is significant potential for the application of AI to help transportation-system operators and owners to advance their goals. Existing applications range from less sophisticated algorithms with simple logic to cutting-edge DL methods. But even the most sophisticated algorithms cannot produce novel insights without well curated data. Our literature review includes a summary of typical AI data sources used in transportation and their challenges and limitations. The research team prepared a Prospectus of AI in transportation to summarize the gathered knowledge (Product 1).

Our data survey revealed a number of data sources that may be used to support TxDOT goals; this effort is centered on applications that may enhance transportation network operations, and integrated corridor management in particular, and emphasis has been placed on data sources that can describe traffic conditions, safety, and the roadway network. There is additional information that would be beneficial from the perspective of corridor management, such as the time and location of planned and unplanned roadway closures and traffic signal timing plans. CTR did not find a standardized source for such data at the time of this report. Samples of the most promising data sources identified in this effort given their availability and coverage are provided in Product 3.

At a later project stage CTR conducted a survey to develop further insights into a strategic approach for data collection. The research team developed a survey because TxDOT personnel are likely to have the best insight on how the organization can progress and where change is most needed. The core goals of the survey were the need to identify high priority applications, to understand internal TxDOT sentiments about current data practices, and to learn about barriers for implementing new methods. The results of the survey have been translated into a number of key insights and recommendations for TxDOT.

The results of this survey and insight from follow-up interviews show that TxDOT has made great strides in using data and pursuing innovative solutions. This progress can be built upon to further progress and help the organization transition into a holistic data strategy. TxDOT will need to expand upon its efforts to formalize data standards, efforts to bring personnel together, and efforts to create contracts that enable innovation.

In order to prioritize use cases for prototyping and select models to be prototyped, the research team, along with TxDOT, explored possible applications of AI in transportation and organized them into the following six categories: system and service planning; asset management; system operations; communication and information; business administration; and public safety and enforcement. An AI strategy that prioritizes use cases that are directly linked to agency goals is helpful for allocating limited resources, driving toward outcomes that are actionable in advancing the agency mission, and gaining leadership support. The applications selected for this effort are based on the priorities identified by the research team after meeting with the project team and following the discussions held during a Workshop organized for this project (P2). Selected use cases for prototyping included:

- Use of Wejo data to understand safety hotspots and gain insights on the impacts of the Pandemic: The team analyzed three weeks of data, two before and one after the pandemic-related shutdown using clustering techniques (unsupervised learning models). Key findings include:
    - Cluster analysis is a useful AI technique for classifying roadway segments based on their characteristics as well as identifying challenging driving environments. Clustering enables TxDOT to readily identify patterns of driving behavior (e.g., areas with high hard braking, high acceleration, and/or speeding events), compare service roads with freeways, and recognize major crash-contributing factors.
    - CV data is a valuable source of information for providing insights into safety, planning, and operations. In particular, event data that has already undergone processing to identify instances of hard braking, hard acceleration, and speeding is extremely useful to a transportation agency. For example, hard braking can identify bottlenecks, stop lights, or queuing from a crash. Hard acceleration can indicate signalized intersections that are close to one another or on-ramps to freeways. This type of information that is packaged in a way to be readily consumable can save a transportation agency significant time and money on computational resources. The data was found to be most useful in conjunction with other data sources, such as roadway geometry and crash reports.

    - Results confirmed observations of lower traffic volumes and higher instances of speeding after stay-at-home orders were issued. Prior to the stay-at-home order, speeding typically occurred outside of downtown on some segments in the northern stretch of the study area. Following the stay-at-home order, speeding events increased throughout the study area and most drastically near downtown.

- Use of Wejo data to understand changes in travel patterns during the pandemic by applying supervised learning techniques: Researchers used several techniques including linear regression, random forest and multilayer perceptron to generate and compare direct demand models before and after the stay-at-home order. The highest percentage reduction in the number of trips was observed at locations with higher service and education opportunities. During lockdown, among the employment categories, only retail employment locations seemed to make a reasonable contribution to the number of trips. Households also made a reduced number of trips during the lockdown. Households that are larger tended to produce more trips than smaller households both before and during the lockdown. Overall, more complex machine learning methods performed better than linear regression when there were outliers in the dataset. Once these outliers were removed, the performance of the linear regression model was similar to that of other machine learning models. The analysis of changes in trip-making patterns after the lockdown is only an example use case of Direct Demand models developed using CV data. More generally, the development and maintenance of such models based on a live stream of CV data will allow for the rapid analysis and identification of changes in travel patterns that occur in any novel situation.

- Real time traffic signal control plans using reinforcement learning models in a simulation environment: Researcher trained reinforced leaning (RL) agents using a simple mesoscopic model and a more complex simulation environment (VISSIM) which allowed to implement timing plans that reflecting protocols used in the field. Experiments suggest that training an RL agent to produce fixed timing plans comparable to those obtained from current approaches may take hundreds of simulated training hours. However, the key advantage of an RL traffic management agent is the ability to respond in near-real time to anomalies and variations in roadway conditions. To illustrate this, we extended our roadway network to include a simulated freeway on which blockages may occur and which will drive significantly higher traffic through the RL agent's intersection. We see from the results that the trained agent is superior in terms of equity, delay, and combined reward in an everyday scenario, and is competitive with the Webster-based timing plan when facing a large obstruction. We also see, however, that in the instances in which the agent must respond to a blocked roadway, there is a significant increase in the variance of all metrics. From this, we observe that,

while an RL agent may take significant training to compete with fixed-time policies, its ability to rapidly respond to changing conditions without human intervention is an asset that warrants further testing in the field. While the increased variance from the trained agent is notable, it has yet to be determined if the variance can be reduced with further training. Nonetheless, the agent has effectively proved to be a suitable agent for managing traffic signal timing plans in simulation, and its practical effects in the field pose an exciting area for future study.

An additional type of model was prototyped to consider the feasibility of providing more accurate travel time information to drivers, in such way that their experienced travel times matched more closely the values displayed in variable message signs. These models relayed in readily available speed and volume data and prototype results were very promising. They were selected for field testing.

- Preliminary models were developed as a proof of concept on I-35 using one year of probe-based speed data and traffic volumes available form smart work zone trailers (which are no longer available). A recurrent neural network (RNN) model was trained for each INRIX segment in order to predict travel times up to one hour into the future at five-minute steps. Total corridor travel time was computed dynamically, by considering arrival time at each successive segment along the route. Results were very promising, with errors almost 40 percent lower than those that would result from using a naïve approach that simply adds the travel time along all segments at the time that the trip starts.

- Models at 4 additional sites were trained for field testing, which involved developing a framework to archive and efficiently access training data, real-time data, model predictions and realized travel times. Limited volume data was available at the selected sites and the results obtained when using it did not show improvement upon models that used only probe-based data. Further research may consider additional analysis of the quality and aggregation of the volume data. Researchers explored different levels of spatial aggregation for model training, seeking to reduce the associated computational effort. Models were trained at the route-level, sub-route level, segment level and pair-of-segments level.

    - Smaller spatial aggregations led to better results in general, which in some cases outperformed the naïve approach significantly, particularly during peak periods. Improvements upon current practice were not as significant as those observed in the preliminary models in all corridors, which suggest that the incorporation of volume data may be critical to maximize the value of these models. Location-specific traffic patterns may also play a role in the potential value of enhanced

travel time prediction methods, which may be more valuable on longer corridors where traffic conditions may change rapidly.

- Models developed to examine differences in spatial aggregation demonstrate benefits of low levels of aggregation. Though the R^2 values of training data are lower for predictions at the lowest time (0.57 for the single segments model versus 0.88 for the entire route) on I-35 SB, when predicting farther in the future the accuracy of predictions is much more stable (0.33 versus 0.04 for the same models).
- Machine learning models also showed benefits when used to provide comparative travel times across routes. In every corridor the percentage of incorrect predictions is less than 10 percent for the single segments model, while the naïve model reaches an almost 15 percent incorrect prediction rate for the Austin SB corridor 30 minutes ahead.
- The naïve model had a MAE in difference in predicted route travel times ranging from 1.11 minutes on the ELP EB corridor to 3.58 minutes on the AUS NB corridor. For the best performing ML model (single segments) the range was smaller, but the values were comparable: 1.47-3.25 minutes on the same corridors. When predicting 30 minutes into the future the naïve model is outperformed substantially by the ML models 1.98-5.63 minutes versus 1.47-2.80 minutes respectively.

Overall, the findings of this project suggest that machine learning holds great promise to improve the analysis, planning an operation of transportation network using emerging data sources. Systematic access and analysis to the data to be considered is critical for implementation, and the implementation of some techniques such as the use of reinforcement learning for signal timing plan optimization may be too complex to implement in-house. Nevertheless, adequate commercial products may become available in the future that may warrant consideration, and further research on the topic is important to further understand potential benefits and limitations. The tools developed in the field-testing phase may be re-deployed at new sites, where they would support a streamlined analysis of data sources in addition to provide access to results from alternative travel time estimation methods and corresponding evaluation. The products developed through this effort are expected to provide TxDOT with a more comprehensive understanding of the potential of machine learning methods and the value of emerging data sources, and to realize additional benefits form existing data sources and subscriptions.

References

Abduljabbar, Rusul, et al. "Applications of artificial intelligence in transport: An overview." *Sustainability* 11.1 (2019): 189.

AIDR. “OUR PARTNERS.” AIDR. Accessed November 27, 2019. http://aidr.qcri.org/.

Arel, Itamar, Cong Liu, Tom Urbanik, and Airton G. Kohls. "Reinforcement learning-based multi-agent system for network traffic signal control." *IET Intelligent Transport Systems* 4, no. 2 (2010): 128-135.

Argility. “Artificial Intelligence: Machine Learning: Deep Learning.” Argility. Accessed November 26, 2019. https://www.argility.com/argility-ecosystem-solutions/iot/machine-learning-deep-learning/.

AutoDesk. “Project Dreamcatcher.” Autodesk Research. Accessed November 27, 2019. https://autodeskresearch.com/projects/Dreamcatcher.

Banerjee, Snehanshu, Mansoureh Jeihani, Nashid K Khadem, and Danny D Brown. 2019. "Units of information on dynamic message signs: a speed pattern analysis." European Transport Research Review 11: 11-15.

Bhavsar, Parth, Ilya Safro, Nidhal Bouaynaya, Robi Polikar, and Dimah Dera. "Machine learning in transportation data analytics." In *Data Analytics for Intelligent Transportation Systems*, pp. 283-307. Elsevier, 2017.

Bispo, Almir, and Carla Andres. “NoSQL Database: The Definitive Guide to NoSQL Databases.” Pandora FMS - The Monitoring Blog, June 25, 2019. https://pandorafms.com/blog/nosql-databases-the-definitive-guide/.

Brownlee, Jason. “Introduction to the Python Deep Learning Library Theano.” Machine Learning Mastery, August 19, 2019. https://machinelearningmastery.com/introduction-python-deep-learning-library-theano/.

Caffe, b. “Interfaces.” Caffe. Accessed November 27, 2019. https://caffe.berkeleyvision.org/tutorial/interfaces.html.

Chen, C., Xiang, H., Qiu, T., et al.: “A rear-end collision prediction scheme based on deep learning in the Internet of vehicles,” J. Parallel Distrib.Comput., 2017, 117, pp. 192–204

Chollet, Francois. “The Future of Deep Learning.” The Keras Blog ATOM, July 18, 2017. https://blog.keras.io/the-future-of-deep-learning.html.

Clark, Jack. “I'll Be Back: The Return of Artificial Intelligence.” Bloomberg.com. Bloomberg, February 3, 2015. https://www.bloomberg.com/news/articles/2015-02-03/i-ll-be-back-the-return-of-artificial-intelligence.

Collins, Katie. “How AI, Twitter and Digital Volunteers Are Transforming Humanitarian Disaster Response.” WIRED. WIRED UK, October 4, 2017. https://www.wired.co.uk/article/digital-humanitarianism.

Cope, Alec. “Jacksonville Debuts High-Tech Streetlights - and They're Watching You.” We Are Change, April 18, 2015. https://wearechange.org/jacksonville-debuts-high-tech-streetlights-and-theyre-watching-you/.

Copeland, Michael. “What’s the Difference Between Artificial Intelligence, Machine Learning, and Deep Learning?” *Nvidia Blog,* (2016). Retrieved from https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Desouza, Kevin C. “Delivering Artificial Intelligence in Government: Challenges and Opportunities.” *Delivering Artificial Intelligence in Government: Challenges and Opportunities*. IBM Center for The Business of Government, 2018.

Ding, C., Cao, X., Liu, C., 2019. How Does the Station-Area Built Environment Influence Metrorail Ridership? Using Gradient Boosting Decision Trees to Identify Non-Linear Thresholds. Journal of Transport Geography 77, 70–78. **https://doi.org/10.1016/j.jtrangeo.2019.04.011**

Federal Highway Administration. 2020. Variable Speed Limits. U.S. Department of Transportation. https://highways.dot.gov/safety/proven-safety-countermeasures/variable-speed-limits.

Firebase. “ML Kit for Firebase | Firebase.” Google. Google. Accessed November 27, 2019. https://firebase.google.com/docs/ml-kit/.

Galang, Jessica. “Miovision Partners with U of T, Launches Lab to Help Cities Make Sense of Traffic Data.” BetaKit, January 9, 2017. https://betakit.com/miovision-partners-with-u-of-t-launches-lab-to-help-cities-make-sense-of-traffic-data/.

Giang, Vivian. “The Growing Business Of Detecting Unconscious Bias.” Fast Company. Fast Company, May 5, 2015. https://www.fastcompany.com/3045899/hit-theground-running/the-growing-business-of-detecting-unconscious-bias.

Greenemeier, Larry. “Human Traffickers Caught on Hidden Internet.” Scientific American, February 8, 2015. https://www.scientificamerican.com/article/human-traffickers-caught-on-hidden-internet/.

Hitachi. “Connected cars will send 25 gigabytes of data to the cloud every hour.” Quartz 2018. Retrieved from https://qz.com/344466/connected-cars-will-send-25-gigabytes-of-data-to-the-cloud-every-hour/

Hodson, Hal. “The AI Boss That Deploys Hong Kong's Subway Engineers.” New Scientist, July 2, 2014. https://www.newscientist.com/article/mg22329764.000-the-ai-boss-that-deploys-hong-kongs-subway-engineers/#.U9efI_ldWSo.

Huang, Jonathan, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer et al. "Speed/accuracy trade-offs for modern convolutional object detectors." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7310-7311. 2017.

Huang, Lei, Weijia Xu, Si Liu, Venktesh Pandey, and Natalia Ruiz Juri. "Enabling versatile analysis of large scale traffic video data with deep learning and HiveQL." In *2017 IEEE International Conference on Big Data (Big Data)*, pp. 1153-1162. IEEE, 2017

Huang, Wenhao, Guojie Song, Haikun Hong, and Kunqing Xie. "Deep architecture for traffic flow prediction: deep belief networks with multitask learning." *IEEE Transactions on Intelligent Transportation Systems* 15, no. 5 (2014): 2191-2201.

IEEE SA. “IEEE SA - Big Data Governance and Metadata Management.” IEEE SA - Big Data Governance and Metadata Management. IEEE SA. Accessed November 26, 2019. https://standards.ieee.org/industry-connections/BDGMM-index.html.

ITS International. “Machine Vision's Transport Offerings Move on Apace.” ITS International - Machine vision's transport offerings move on apace. ITS International, June 2016.

http://www.itsinternational.com/categories/detection-monitoring-machine-vision/features/machine-visions-transport-offerings-move-on-apace/.

ITS JPO. “Enterprise Data.” Intelligent Transportation Systems - Enterprise Data. ITS JPO: US DOT. Accessed November 26, 2019. https://www.its.dot.gov/research_areas/enterprise.htm

Jahangiri, Arash, and Hesham A. Rakha. "Applying machine learning techniques to transportation mode recognition using mobile phone sensor data." *IEEE transactions on intelligent transportation systems* 16, no. 5 (2015): 2406-2417.

Jain, Kunal. “Scikit-Learn In Python - the Most Important Machine Learning Tool I Learnt Last Year.” Analytics Vidhya, January 5, 2015. https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/.

Jin, F., Sun, S.: ‘Neural network multitask learning for traffic flowforecasting’. IEEE Int. Joint Conf. Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1June 2008, pp. 1897–1901

Jindahra, Pavitra, and Kasem Choocharukul. 2013. "Short-Run Route Diversion: An Empirical Investigation into Variable Message Sign Design and Policy Experiments." IEEE Transactions on Intelligent Transportation Systems 14 (1): 388-397.

Júnior, Jair Ferreira, Eduardo Carvalho, Bruno V. Ferreira, Cleidson de Souza, Yoshihiko Suhara, Alex Pentland, and Gustavo Pessin. "Driver behavior profiling: An investigation with different smartphone sensors and machine learning." *PLoS one* 12, no. 4 (2017): e0174959.
Keras. “Keras: The Python Deep Learning Library.” Home - Keras Documentation. Accessed November 27, 2019. https://keras.io/.

Knight, Will. “How Artificial Intelligence Can Fight Air Pollution in China.” MIT Technology Review. MIT Technology Review, September 1, 2015. https://www.technologyreview.com/s/540806/how-artificial-intelligence-can-fight-air-pollution-in-china/.

Koesdwiady, Arief, Ridha Soua, and Fakhreddine Karray. "Improving traffic flow prediction with weather information in connected cars: A deep learning approach." *IEEE Transactions on Vehicular Technology* 65, no. 12 (2016): 9508-9517.

Liang, Xiaoyuan, et al. "Deep reinforcement learning for traffic light control in vehicular networks." *arXiv preprint arXiv:1803.11115* (2018).

Liang, Xiaoyuan, Xunsheng Du, Guiling Wang, and Zhu Han. "Deep reinforcement learning for traffic light control in vehicular networks." *arXiv preprint arXiv:1803.11115* (2018).

Ma, Xiaolei, Haiyang Yu, Yunpeng Wang, and Yinhai Wang. "Large-scale transportation network congestion evolution prediction using deep learning theory." *PloS one* 10, no. 3 (2015): e0119044.

Manzoni, Vincenzo, Diego Maniloff, Kristian Kloeckl, and Carlo Ratti. "Transportation mode identification and real-time CO2 emission estimation using smartphones." *SENSEable City Lab, Massachusetts Institute of Technology, nd* (2010).

Marr, Bernard. “How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read.” Forbes, Forbes Magazine, 9 July 2018, www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#660b219960ba.

Marshall, Aarian. “Parking a Truck Is a Pain in the Butt. Tech to the Rescue!” Wired. Conde Nast, June 3, 2017. https://www.wired.com/2016/06/parking-truck-pain-butt-tech-rescue/.

Matías, J. M., J. Taboada, C. Ordóñez, and P. G. Nieto. "Machine learning techniques applied to the determination of road suitability for the transportation of dangerous substances." Journal of hazardous materials 147, no. 1-2 (2007): 60-66.

Mehndiratta, Shomik, and Bernardo Alvim. “Big Data Comes to Transport Planning: How Your Mobile Phone Helps Plan That Rail Line.” World Bank Blogs. World Bank, December 30, 2014. http://blogs.worldbank.org/transport/big-data-comes-transport-planning-how-your-mobile-phone-helps-plan-rail-line.

Mehr, Hila. “Artificial Intelligence for Citizen Services and Government.” *Artificial Intelligence for Citizen Services and Government*, 2017.

Metz, Rachel. “App Listens for Danger When You're Not Paying Attention.” MIT Technology Review. MIT Technology Review, September 19, 2014. https://www.technologyreview.com/s/524971/app-listens-for-danger-when-youre-not-paying-attention/.

Moren, Dan. “Autonomous Robots To Help Remove Car Bomb Threats.” Popular Science. Popular Science, March 18, 2019. https://www.popsci.com/autonomous-robots-help-remove-car-bomb-threats/.

Morris, David Z. “How AT&T Is Using Drivers' Cellular Data to Help Fix California Traffic.” Fortune. Fortune, October 16, 2015. http://fortune.com/2015/10/16/att-using-big-data-to-fix-traffic/.

Nandurge, Priyanka A., and Nagaraj V. Dharwadkar. "Analyzing road accident data using machine learning paradigms." *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 2017

Nguyen, Hoang, et al. "Deep learning methods in transportation domain: a review." *IET Intelligent Transport Systems* 12.9 (2018): 998-1004.

NYSDOT Office of Traffic Safety and Mobility. 2018. "Variable Message Sign Guidelines." December. https://www.dot.ny.gov/divisions/operating/oom/transportation-systems/repository/VMS%20%20Guidelines%20December%202018%20-%20FINAL.pdf.

Olmez, Sedar, Liam Douglas-Mann, Ed Manley, Keiran Suchak, Alison Heppenstall, Dan Birks, and Annabel Whipp. 2021. "Exploring the Impact of Driver Adherence to Speed Limits and the Interdependence of Roadside Collisions in an Urban Environment: An Agent-Based Modelling Approach." Applied Sciences 11 (12).
Ortúzar, J. de D., Willumsen, L.G., 2011. Modelling Transport, 4th edition. ed. Wiley, Chichester, West Sussex, United Kingdom.
Popper, Ben. “The Smart Bots Are Coming and This One Is Brilliant.” The Verge, April 7, 2016. https://www.theverge.com/2016/4/7/11380470/amy-personal-digital-assistant-bot-ai-conversational.

Progress. “Cognitive Predictive Maintenance Platform for Industrial IoT - Progress DataRPM.” Progress.com. Accessed November 27, 2019. http://www.datarpm.com/offerings.php.

Reason, James, Antony Manstead, Stephen Stradling, James Baxter, and Karen Campbell. 1990. "Errors and violations on the roads: A real distinction?" Ergonomics 33 (10-11): 1315-1332.

Reinolsmann, Nora, Wael Alhajyaseen, Tom Brijs, Ali Pirdavani, Veerle Ross, Qinaat Hussain, and Kris Brijs. 2022. "Delay or travel time information? The impact of advanced travel

information systems on drivers' behavior before freeway work zones." Transportation Research Part F: Traffic Psychology and Behavior 87: 454-476.

Rodrigues, Filipe, Pereira, Francisco & Ben-Akiva, Moshe. “Using Data from the Web to Predict Public Transport Arrivals Under Special Events Scenarios.” Journal of Intelligent Transportation Systems (2013): 19. 10.1080/15472450.2013.868284.

Schatsky, David, Craig Muraskin, and Ragu Gurumurthy. “Cognitive Technologies: The Real Opportunities for Business.” Deloitte Insights. Deloitte. Accessed November 26, 2019. https://www2.deloitte.com/insights/us/en/deloitte-review/issue-16/cognitive-technologies-business-applications.html#endnote-15.

Shelhamer, Evan, Jeff Donahue, Jon Long, Yangqing Jia, and Ross Girshick. “DIY Deep Learning for Vision: a Hands-On Tutorial with Caffe.” Google Slides. Google. Accessed November 27, 2019. https://docs.google.com/presentation/d/1UeKXVgRvvxg9OUdh_UiC5G71UMscNPlvArsWER41PsU/edit#slide=id.gc2fcdcce7_216_349.

Shetty, Sunith. “What Is PyTorch and How Does It Work?” Packt Hub, September 20, 2018. https://hub.packtpub.com/what-is-pytorch-and-how-does-it-work/.

Soriguera, Francesc. 2014. "On the value of highway travel time information systems." Transportation Research Part A: Policy and Practice 70: 294-310.

Stack, Tim. “Internet of Things (IoT) Data Continues to Explode Exponentially. Who Is Using That Data and How?” Cisco Blog, 2018.

Tepper, Fitz. “Lawyaw Uses AI to Help Lawyers Draft Documents Faster.” TechCrunch. TechCrunch, March 23, 2018. https://techcrunch.com/2018/03/23/lawyaw-uses-ai-to-help-lawyers-draft-documents-faster/.

Theano. “Welcome.” Welcome - Theano 1.0.0 documentation. Accessed November 27, 2019. http://deeplearning.net/software/theano/.

Tsai, Yichang, Zhaohua Wang, and Chengbo Ai. “Implementation of Automatic Sign Inventory and Pavement Condition Evaluation on Georgia's Interstate Highways.” *Implementation of Automatic Sign Inventory and Pavement Condition Evaluation on Georgia's Interstate Highways*. Georgia Department of Transportation, February 2017.

https://ntlrepository.blob.core.windows.net/lib/60000/60900/60954/15-11.pdf.

Ullman, Jerry. 2013. Construction Traveler Information System for I-35 Widening in Central Texas. Bettendorf, Iowa: Federal Highway Administration.
US Census Bureau, 2021. About the ACS [WWW Document]. The United States Census Bureau. URL https://www.census.gov/programs-surveys/acs/about.html (accessed 1.24.21).
Veneziano, David, Larry Hayden, and Jared Ye. 2010. Effective Deployment of Radar Speed Signs. Federal Highway Administration.

Wang, Shenhao, and Jinhua Zhao. "Framing discrete choice model as deep neural network with utility interpretation." *arXiv preprint arXiv:1810.10465* (2018).

Washington State Department of Transportation. n.d. Variable message signs. https://tsmowa.org/category/intelligent-transportation-systems/variable-message-signs.
Winter, Kathy. "For Self-Driving Cars, There's Big Meaning Behind One Big Number: 4 Terabytes" Intel 2017. Retrieved from: *https://newsroom.intel.com/editorials/self-driving-cars-big-meaning-behind-one-number-4-terabytes/*

Wu, Yuankai, Huachun Tan, Lingqiao Qin, Bin Ran, and Zhuxi Jiang. "A hybrid deep learning based traffic flow prediction method and its understanding." *Transportation Research Part C: Emerging Technologies* 90 (2018): 166-180.
Yan, X., Liu, X., Zhao, X., 2020. Using machine learning for direct demand modeling of ridesourcing services in Chicago. Journal of Transport Geography 83, 102661. https://doi.org/10.1016/j.jtrangeo.2020.102661126.
Yang, Jidong J., Ying Wang, and Chih-Cheng Hung. *Monitoring and Assessing Traffic Safety at Signalized Intersections Using Live Video Images*. No. RP 14-29. 2018.

Yao, Huaxiu, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. "Deep multi-view spatial-temporal network for taxi demand prediction." In Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

Yegulalp, Serdar. "What Is TensorFlow? The Machine Learning Library Explained." InfoWorld. InfoWorld, June 18, 2019. https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html.

Yu, Yun, Xiao Han, Bin Jia, Rui Jiang, Zi-You Gao, and H Michael Zhang. 2021. "Is providing inaccurate pre-trip information better than providing no information in the morning commute under stochastic bottleneck capacity?" Transportation Research Part C: Emerging Technologies.

Zhang, Shen, Jinjun Tang, Hua Wang, and Yinhai Wang. "Enhancing traffic incident detection by using spatial point pattern analysis on social media." Transportation Research Record 2528, no. 1 (2015): 69-77.

Zineddin, Abdul, Shauna Hallmark, Omar Smadi, and Neal Hawkins. 2016. Spotlighting Speed Feedback Signs. Federal Highway Administration.

Appendix A. Survey Results

Note: Results are shown for the 25 respondents who fully completed the survey. Input from any partial completions is not included. Contact information has been omitted for anonymity purposes.

Q1: This survey is part of a TxDOT RTI project where our research group is developing a data acquisition strategy for TxDOT. It includes prioritizing potential uses for new data sources and artificial intelligence (AI) tools. The goal of this survey is to understand what applications are most relevant to TxDOT in the near future (one to three years), so that TxDOT can prioritize their efforts. All results will be aggregated and your input will be anonymous. This survey will take roughly five to ten minutes to complete. We appreciate your input and value your thoughtful participation.

Q2: Please provide your contact information (name, organization, and email)

Q3: Do you have experience working with data (acquiring, analyzing, managing) or analysis tools to improve transportation planning, maintenance, operations, etc.?

Results:

Choice	Percentage (%)	Count
Yes, I have significant experience and knowledge in this area.	48.00	12
Yes, I have some experience and limited knowledge.	40.00	10
No, I have no experience but I am still knowledgeable on the subject.	0.00	0
No, I have neither experience nor knowledge on this subject.	12.00	3

Q4: We have listed eight areas where new data sources or AI tools/services could be used to enhance operations, safety, and planning. Please select three which you consider to be highest priority for TxDOT:

Results:

Choice	Percentage(%)	Count
Incident Detection – Detect incidents faster and more accurately in order to aid emergency responders (improve incident response/clearance times).	13.33	10
Asset Management – Improve collection and accuracy of road condition data. Increase the availability of objective metrics, translate data into actions, and prioritize maintenance activities.	12.00	9
Truck Parking Availability – Provide truck drivers with predictive estimates of parking availability to reduce truck parking challenges.	2.67	2
Flood Detection – Detect flooding events earlier where flood sensors are not installed and improved travel recommendations during flooding events.	6.67	5
Integrated Corridor Management – Implement adaptive signal timing which improves throughput at signalized intersections and adjusts quickly during unexpected periods of congestion.	18.67	14
Identify Safety Hotspots – Leverage new datasets in tandem with existing resources (CRIS, intersection geometry, etc.) to help planners locate and address road segments with a high concentration of unsafe behaviors and collisions.	25.33	19
Disaster Response – Leverage social media posts for improved awareness of conditions, better understanding of community needs, and vast information input at a low cost.	8.00	6
Travel Trends & Forecasting – Use data from connected cars to understand travel behavior trends and connect this with other data to forecast expected future travel demand and behavior.	13.33	10

Q5: Besides the eight applications in the previous questions, are there other AI/data applications that you would like TxDOT to explore?

Results:

1. travel time prediction
2. TP&D -data standards for developing projects, comparing it to when projects are in construction -how 'well' the project did, and in the future, if you are developing a project, how likely are you to have a change order, or addendum, etc.
3. Optimization of asset management resources to get the biggest benefit for the available resources.
4. Project Management/ROW and Utilities Sections
5. Data management and data visualization
6. Don't know of any.
7. Not really sure
8. Similar to Asset management: Utility corridor management. Better ways to track and lookup utility lines in our ROW for ease of design and prevent delay caused by surprise conflicts in the field.
9. Factors leading to infrastructure projects to be on time and on budget. Identify and predict to help w/ future outcomes from there.

Q6: Now, rank your top three applications from highest priority (top) to lowest priority (bottom):

Results:

Field	Rank 1 %	Rank 1 Count	Rank 2 %	Rank 2 Count	Rank 3 %	Rank 3 Count
Incident Detection	*25.00*	2	*50.00*	4	*25.00*	2
Asset Management	*14.29*	1	*42.86*	3	*42.86*	3
Truck Parking Availability	*0.00*	0	*50.00*	1	*50.00*	1
Flood Detection	*0.00*	0	*33.33*	1	*66.67*	2
Integrated Corridor Management	*53.85*	7	*7.69*	1	*38.46*	5
Identify Safety Hotspots	*46.67*	7	*33.33*	5	*20.00*	3
Disaster Response	*20.00*	1	*40.00*	2	*40.00*	2
Travel Trends & Forecasting	*28.57*	2	*42.86*	3	*28.57*	2

Q7: When selecting and ranking applications, select up to three of the following that drove your decision:

Results:

Choice	Percentage (%)	Count
Readiness: could be implemented quickly with existing technology	11.59	8
Alignment: fits well with existing priorities and other projects	24.64	17
Impact: could create a noticeable improvement	27.54	19
Resilience: would improve reliability during unexpected conditions	13.04	9
Novelty: innovative approach unlike any existing strategies	2.90	2
Funding: feasibility of acquiring or allocating funds	4.35	3
Sustainability: continued return on investment over time	14.49	10
Other: "Efficient: funneling data sets, which enables insights, is more efficient process of what already occurs"	1.45	1

Q8: In terms of their use of new data sources and AI tools, TxDOT is…

Results:

Choice	Percentage (%)	Count
Ahead of most states.	12.00	3
On par with most states.	28.00	7
Behind most states.	4.00	1
I'm not sure.	56.00	14

Q9: From this list, please identify up to two major barriers you experience when working to implement a new data source or analysis tool:

Results:

Choice	Percentage (%)	Count
Funding, lack of available funds for this purpose.	17.39	8
Skillset, our team does not have the expertise needed.	21.74	10
Awareness, limited knowledge of existing tools and data sources.	19.57	9
Authority, I am not in a position to make this decision.	15.22	7
Priority, there are other initiatives that are more pressing.	10.87	5
Unsure, I have not tried to implement something like this.	4.35	2
Not applicable, I do not experience significant barriers in this process.	0.00	0
Other: please specify.	10.87	5

Text Entries from Other:

1. Federal and state rules, policy and guidance direct environmental analysis. We are often required to conduct worst case analysis that neither factors nor needs new and emerging datasets. It often requires projections that are also worst case (e.g. assume little to no market penetration overtime for electric vehicles).
2. Data Sets and Standards do not necessarily exist; conflict -different teams have different ideas
3. Time; We are understaffed.
4. Administration does not understand the technical concepts to make it happen, and therefore avoid going after these solutions with currently available tools and skillsets.
5. IT coordination, or lack thereof

Q10: When evaluating a new data source and/or new analysis tool for acquisition, select up to three of the following that would drive your decision:

Results:

Choice	Percentage (%)	Count
Cost: Preference for the lowest cost solution available.	1.37	1
Data Format: Consideration of whether data is delivered raw or processed.	4.11	3
Endorsement: An active community of practice or other form of division support to understand the tool.	13.70	10
Visualizations: Ability of the product to translate results into key metrics and insights.	23.29	17
Maturity: Age and experience of the company that is offering the product	4.11	3
Pilot: A desire to test the benefit to TxDOT, District Operations or the Traveling Public	12.33	9
Evidence: Successful use cases from other jurisdictions.	23.29	17
Evaluation: A cost/benefit analysis which weighs in favor of the source/tool.	9.59	7
Data Granularity: Preference for data that has low latency and high frequency.	8.22	6

Q11: Is there any additional information or context that you think is important to include in a data acquisition strategy for TxDOT?

Results:

1. No
2. Needs to have a holistic vision in terms of data needs
3. Conflicts occur in Division, which presents different messages to districts. What message is presented to districts should be an enterprise strategy.
4. Look at the big picture and realize where we get the biggest impact. Go after those data sources and use them to create improvement.
5. N/A
6. Some amount of training from actual trainers. Some of the training is not helpful.
7. It must be paired with data governance. Otherwise there is no agreement on how to use the tool or data.

8. Yes, slightly more centralization of data acquisition and consideration for how governance will work. A central data warehouse continues to be a huge need for the agency, which would lead to more use cases being discovered.
9. Use cases. Not too sure what the applications for my department will be, but I know that there will be some good applications for my department
10. Speed of implementation. Some programs were announced years ago, yet to be implemented, and enthusiasm has waned.
11. Goal alignment and how this new data/tool fits in w/ existing data and tools.

Q12: Thank you for providing your input. We would like to reach out to a handful of respondents to discuss their input and the overall results. Would you be interested in a 30-minute follow-up meeting?

Results:

Choice	Percentage (%)	Count
Yes	33.33	8
No	66.67	16

Appendix B. Dynamic Travel Times

Consider a freeway corridor that has been divided into segments . We assume that the travel time on each on these segments changes as a function of time, which is discretized in time intervals . In this context represents the travel time on segment at time step . We define the instantaneous travel time on corridor at time as. The dynamic travel time is defined recursively as

Where is the partial dynamic travel time up to segment s-1. We assume that for s=1.

Figure 111 exemplifies the calculation of *ITT* and *DTT* for a hypothetical freeway corridor *S* consisting of 4 segments *S*={1,2,3,4} assuming that travel times are expressed in terms of time intervals (e.g. corresponds to a travel time equal to 3 time steps).

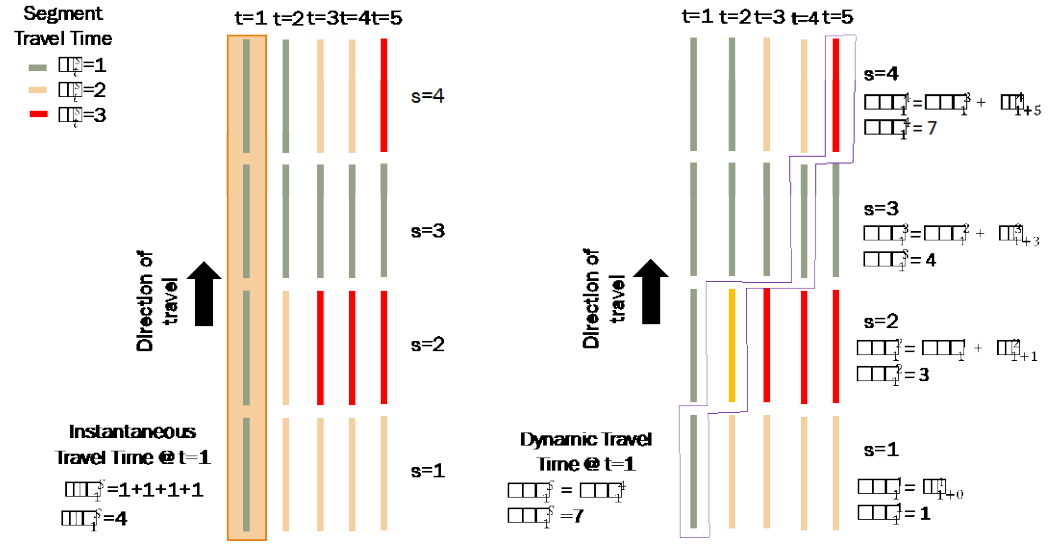
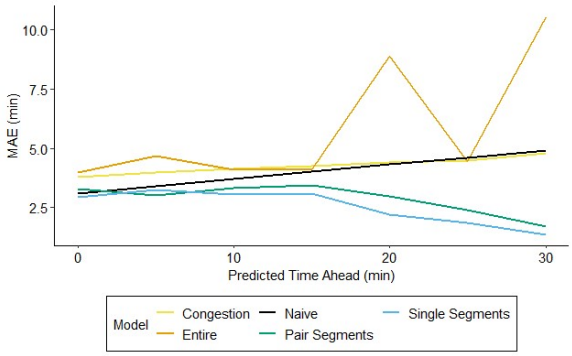
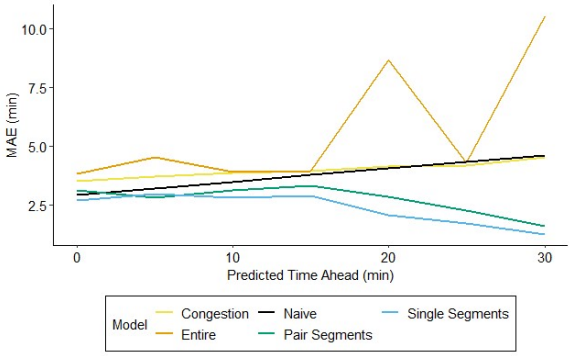


Figure 111. Example of Instantaneous and dynamic travel time computation.

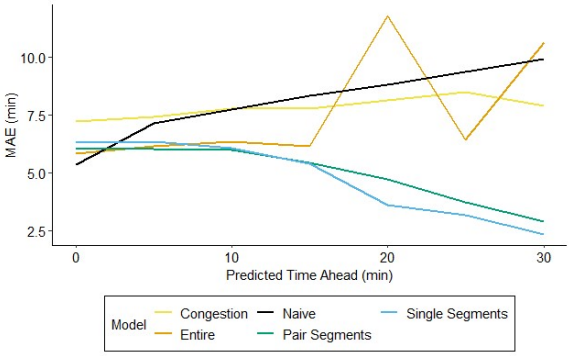
Appendix C. Implementation Results of Phase II Models



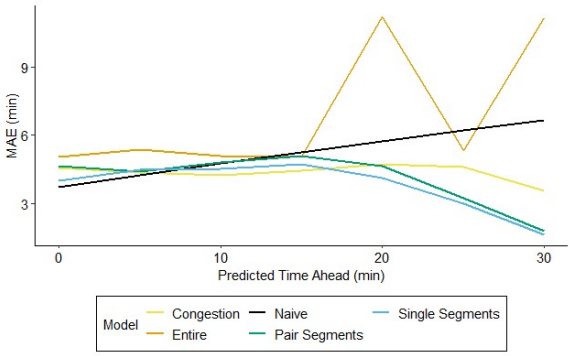
a) I-35 SB average real-time MAE (min).



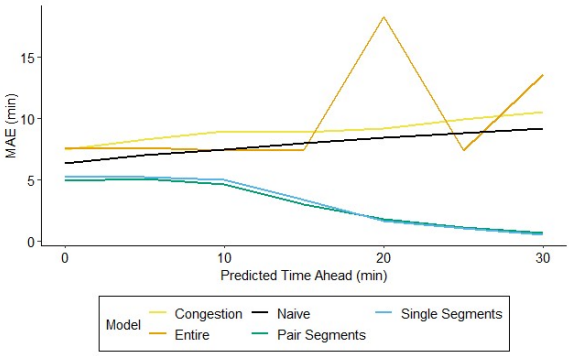
b) I-35 SB real-time MAE (min) for typical conditions.



c) I-35 SB real-time MAE (min) for atypical conditions.

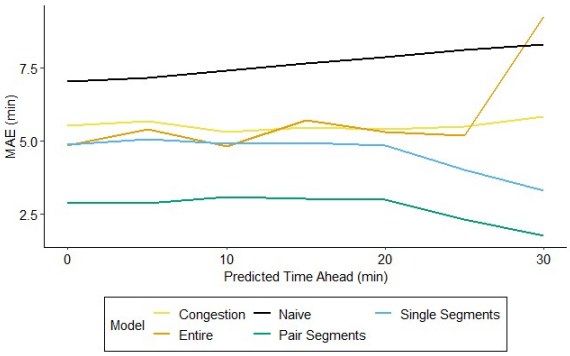


d) I-35 SB real-time MAE (min) for the AM peak period.

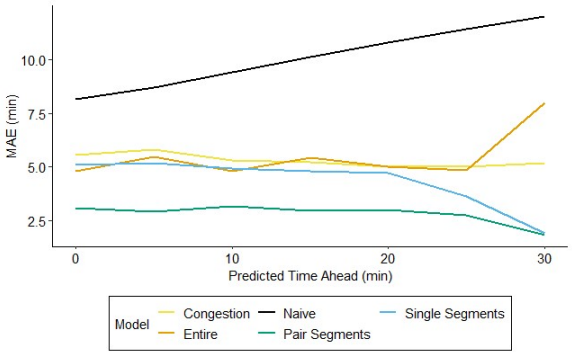


e) I-35 SB real-time MAE (min) for the PM peak period.

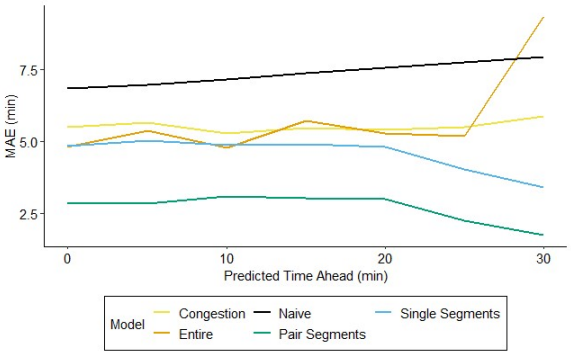
Figure 112: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 SB.



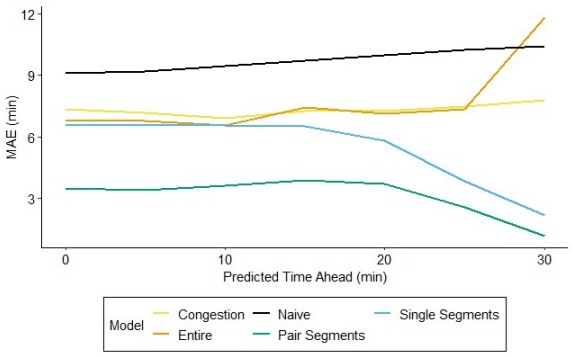
a) I-35 NB average real-time MAE (min).



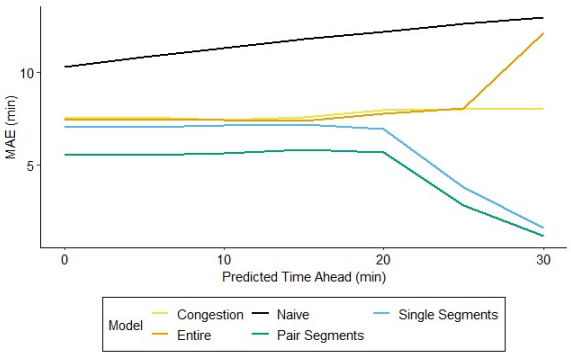
b) I-35 NB real-time MAE (min) for typical conditions.



c) I-35 NB real-time MAE (min) for atypical conditions.

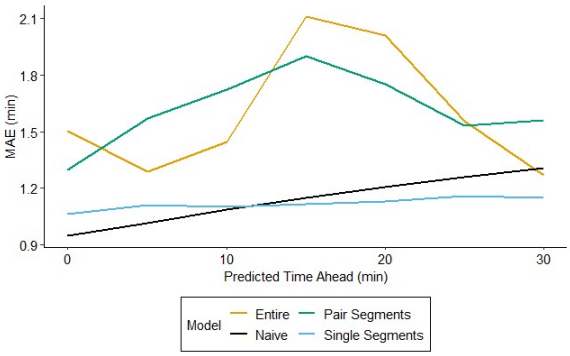


d) I-35 NB real-time MAE (min) for the AM peak period.

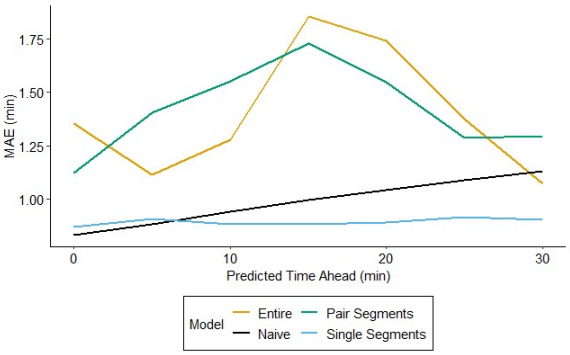


e) I-35 NB real-time MAE (min) for the PM peak period.

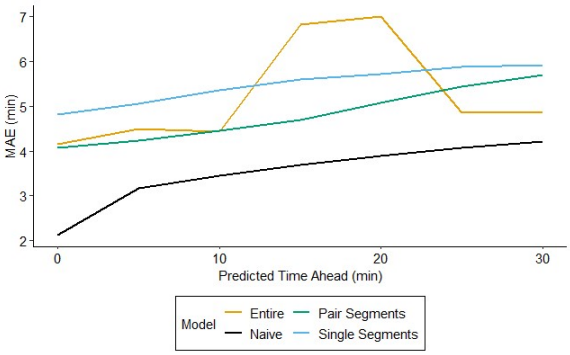
Figure 113: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 NB.



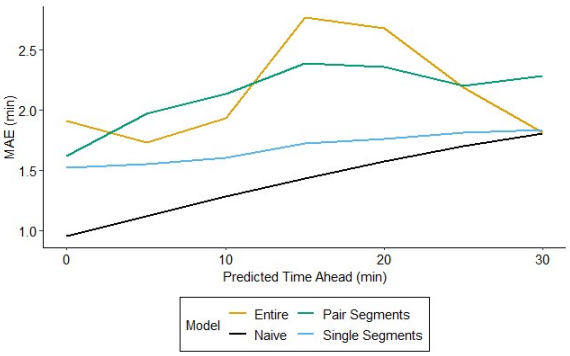
a) SH 130 SB average real-time MAE (min).



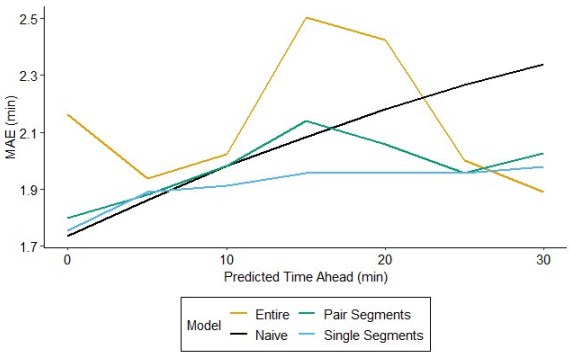
b) SH 130 SB real-time MAE (min) for typical conditions.



c) SH 130 SB real-time MAE (min) for atypical conditions.

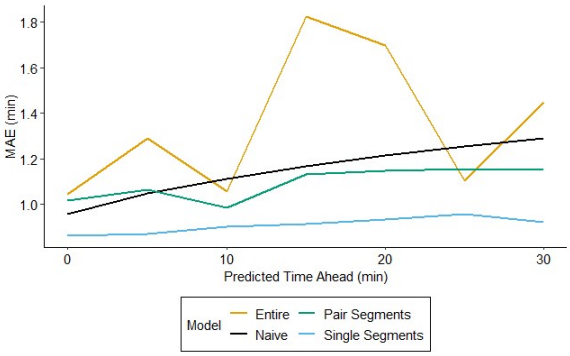


d) SH 130 SB real-time MAE (min) for the AM peak period.

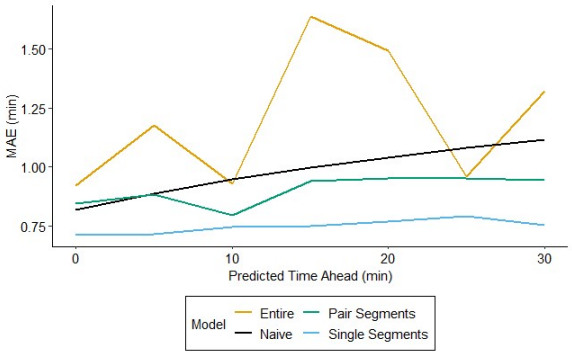


e) SH 130 SB real-time MAE (min) for the PM peak period.

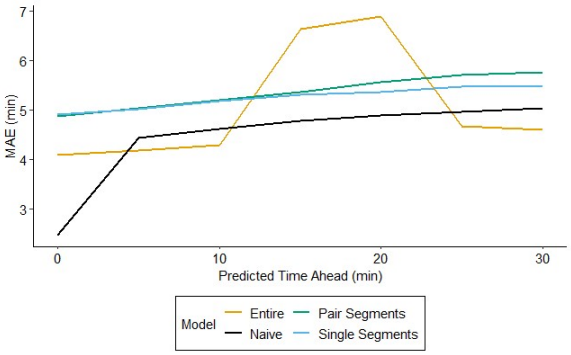
Figure 114: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 SB.



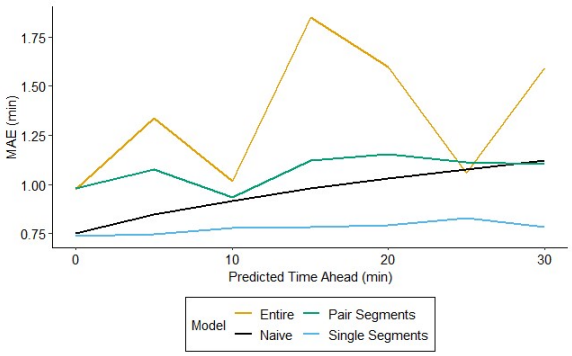
a) SH 130 NB average real-time MAE (min).



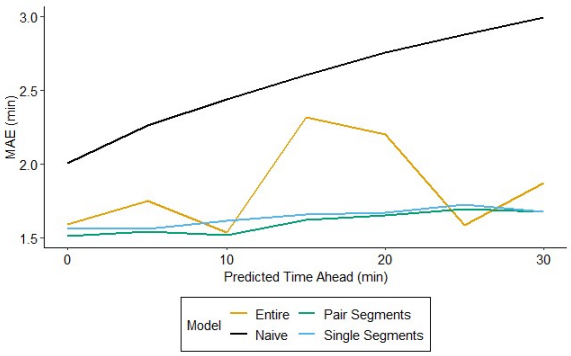
b) SH 130 NB real-time MAE (min) for typical conditions.



c) SH 130 NB real-time MAE (min) for atypical conditions.

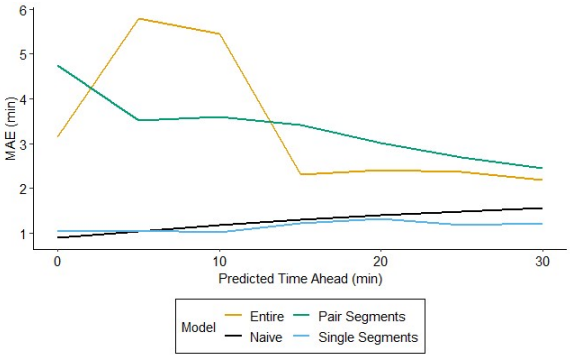


d) SH 130 NB real-time MAE (min) for the AM peak period.

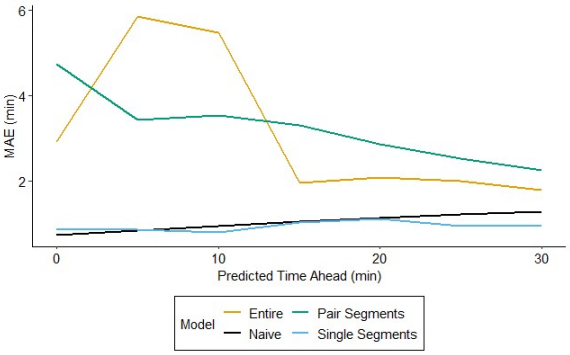


e) SH 130 NB real-time MAE (min) for the PM peak period.

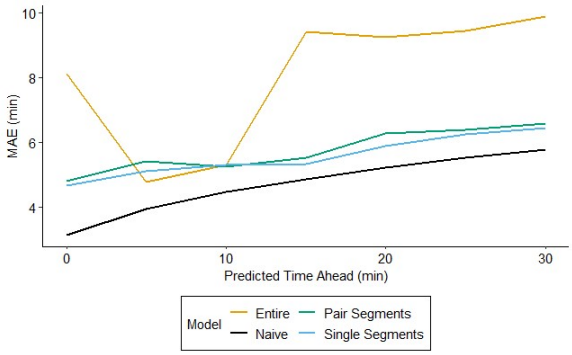
Figure 115: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 NB.



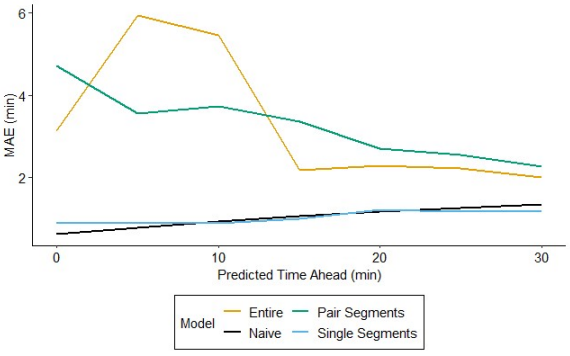
a) I-10 EB average real-time MAE (min).



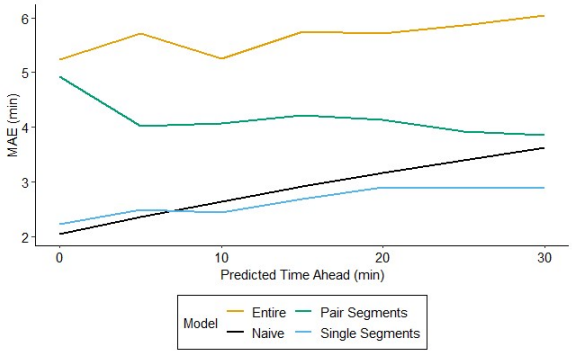
b) I-10 EB real-time MAE (min) for typical conditions.



c) I-10 EB real-time MAE (min) for atypical conditions.

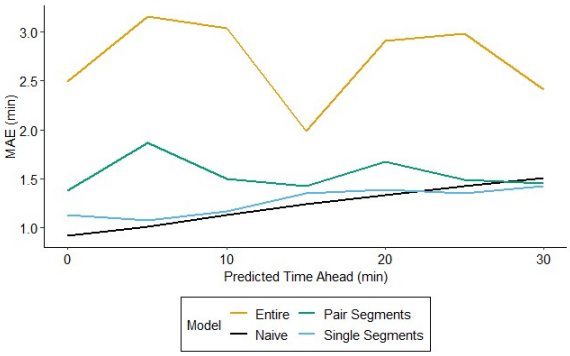


d) I-10 EB real-time MAE (min) for the AM peak period.

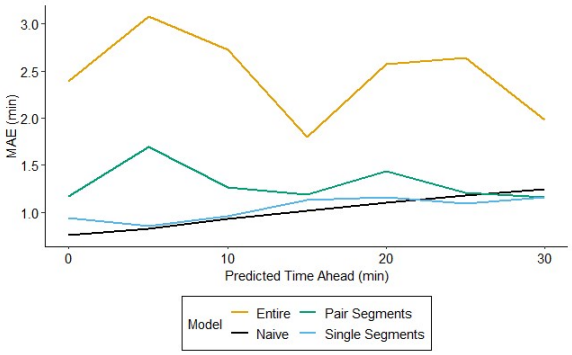


e) I-10 EB real-time MAE (min) for the PM peak period.

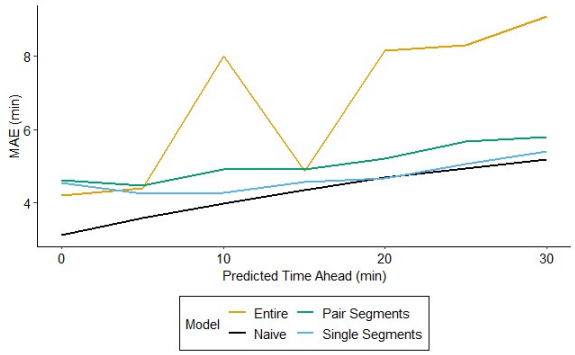
Figure 116: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 EB.



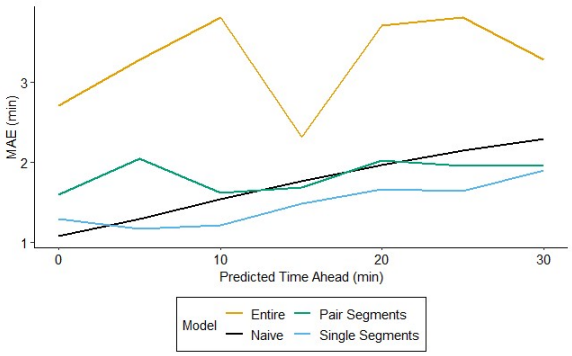
a) I-10 WB average real-time MAE (min).



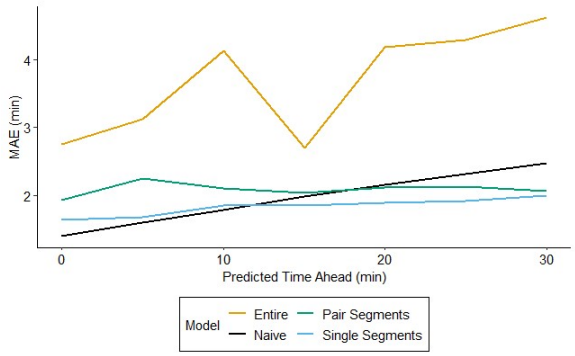
b) I-10 WB real-time MAE (min) for typical conditions.



c) I-10 WB real-time MAE (min) for atypical conditions.

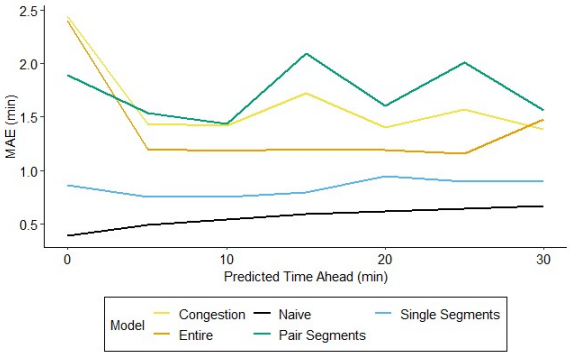


d) I-10 WB real-time MAE (min) for the AM peak period.

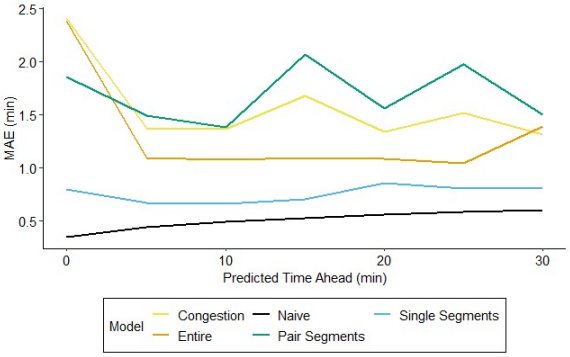


e) I-10 WB real-time MAE (min) for the PM peak period.

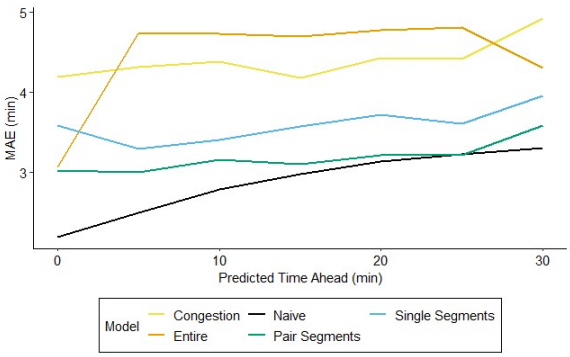
Figure 117: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 WB.



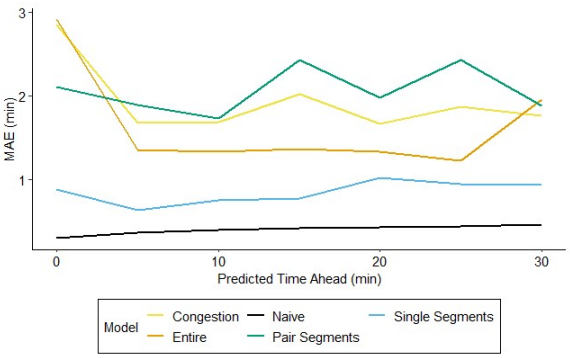
a) SL 375 EB average real-time MAE (min).



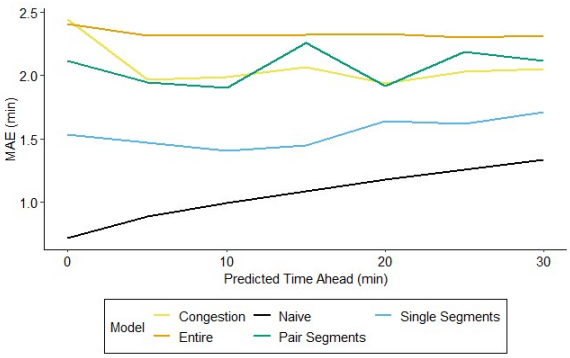
b) SL 375 EB real-time MAE (min) for typical conditions.



c) SL 375 EB real-time MAE (min) for atypical conditions.

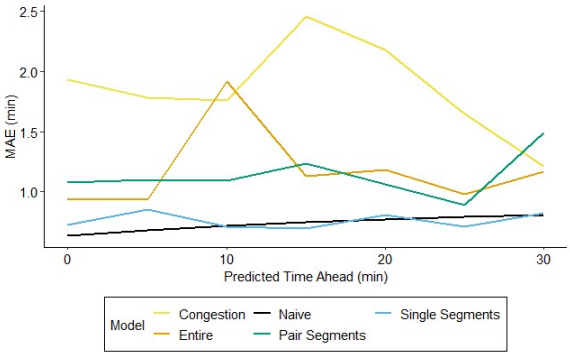


d) SL 375 EB real-time MAE (min) for the AM peak period.

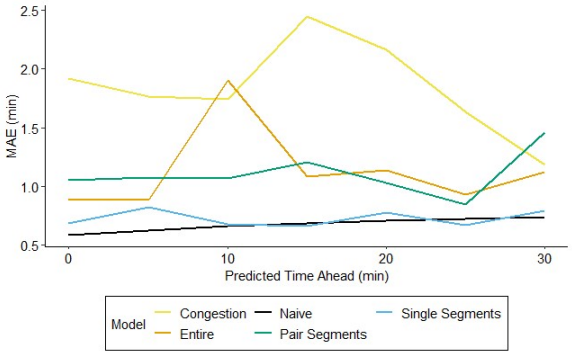


e) SL 375 EB real-time MAE (min) for the PM peak period.

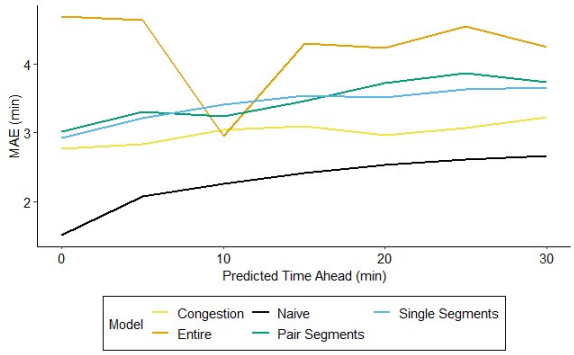
Figure 118: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 EB.



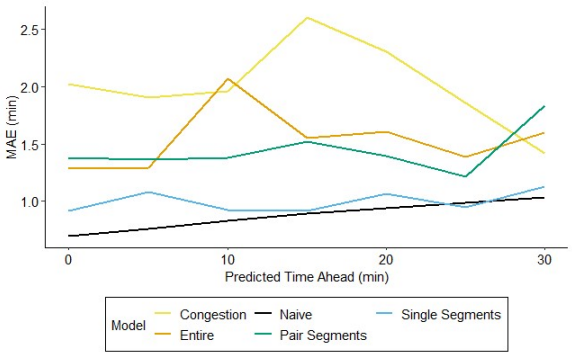
a) SL 375 WB average real-time MAE (min).



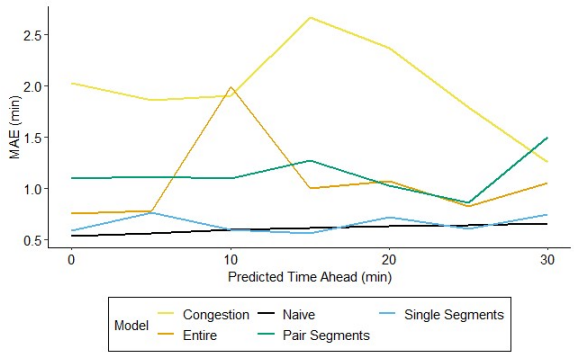
b) SL 375 WB real-time MAE (min) for typical conditions.



c) SL 375 WB real-time MAE (min) for atypical conditions.

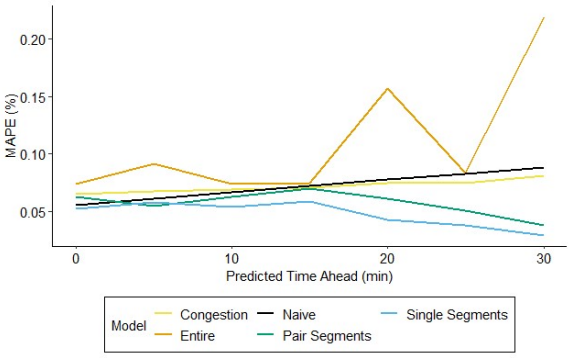


d) SL 375 WB real-time MAE (min) for the AM peak period.

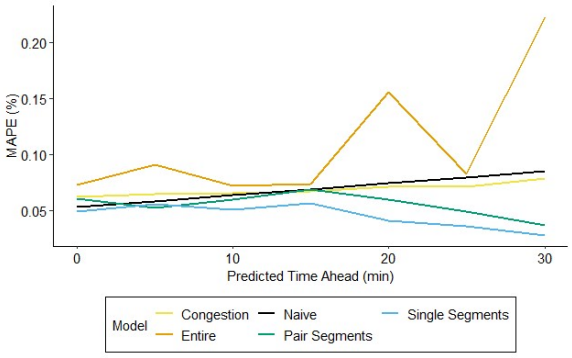


e) SL 375 WB real-time MAE (min) for the PM peak period.

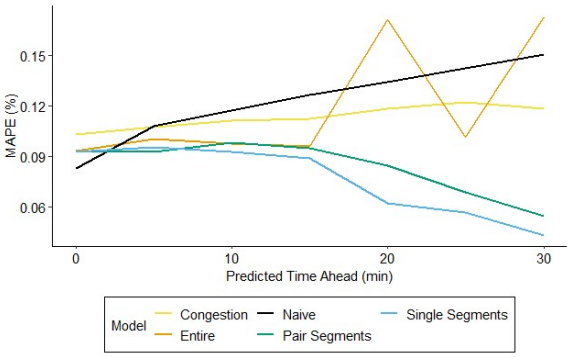
Figure 119: MAE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 WB.



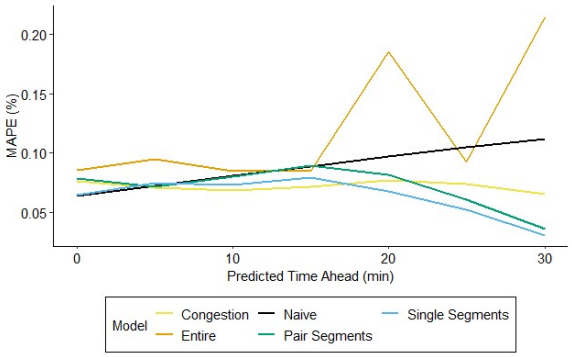
a) I-35 SB average real-time MAPE (%).



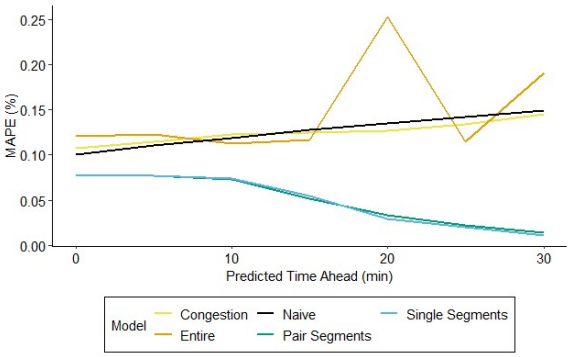
b) I-35 SB real-time MAPE (%) for typical conditions.



c) I-35 SB real-time MAPE (%) for atypical conditions.

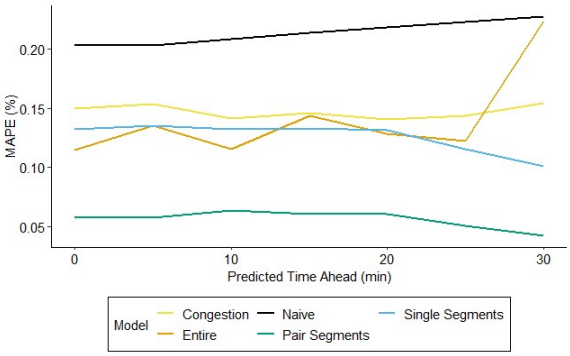


d) I-35 SB real-time MAPE (%) for the AM peak period.

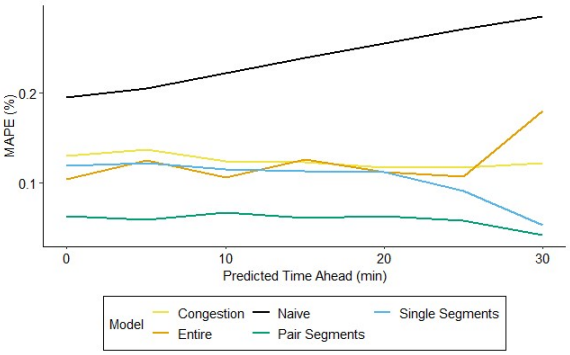


e) I-35 SB real-time MAPE (%) for the PM peak period.

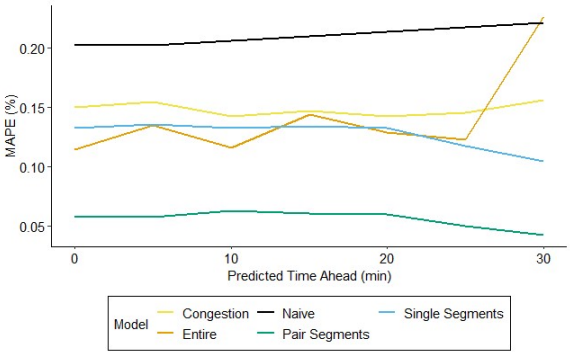
Figure 120: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 SB.



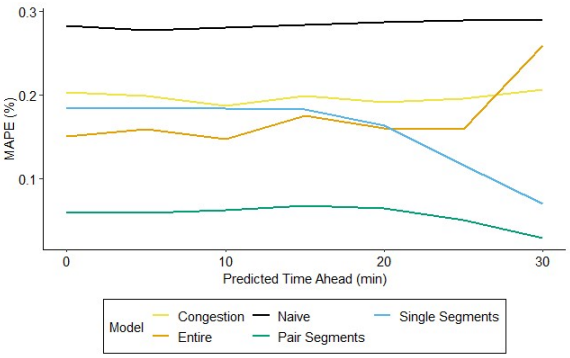
a) I-35 NB average real-time MAPE (%).



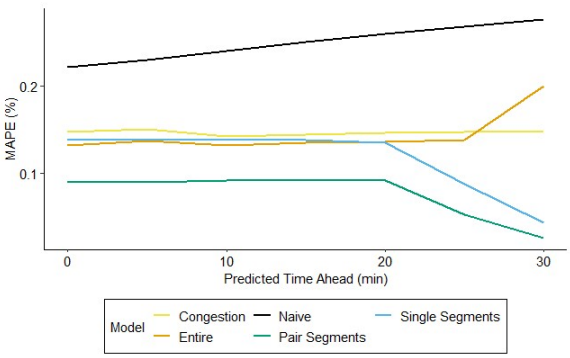
b) I-35 NB real-time MAPE (%) for typical conditions.



c) I-35 NB real-time MAPE (%) for atypical conditions.

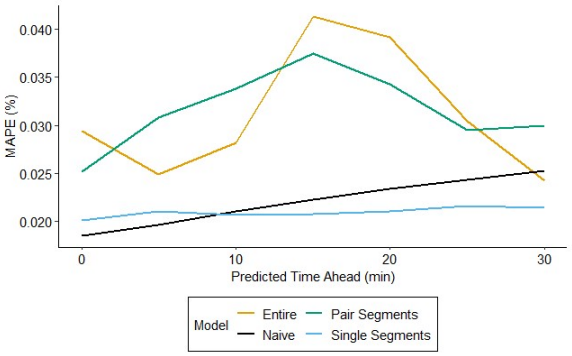


d) I-35 NB real-time MAPE (%) for the AM peak period.

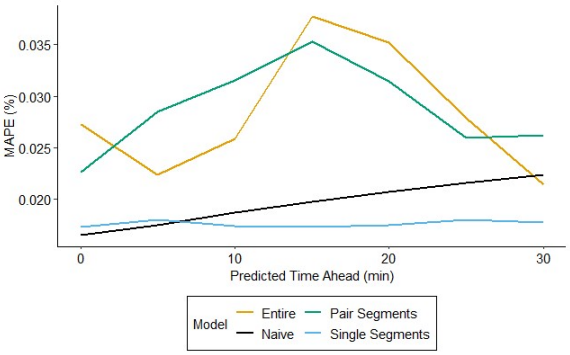


e) I-35 NB real-time MAPE (%) for the PM peak period.

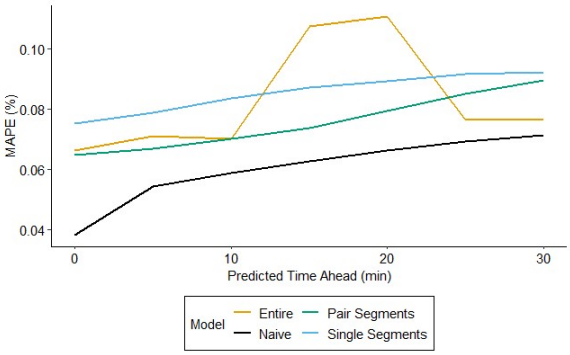
Figure 121: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 NB.



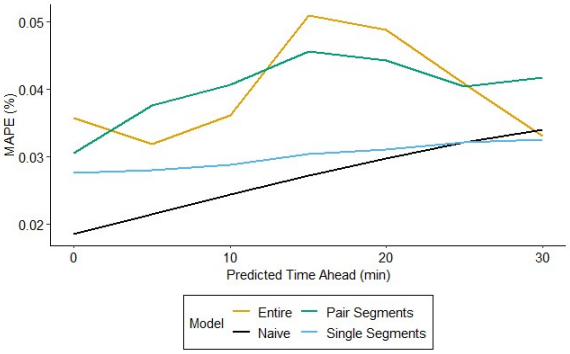
a) SH 130 SB average real-time MAPE (%).



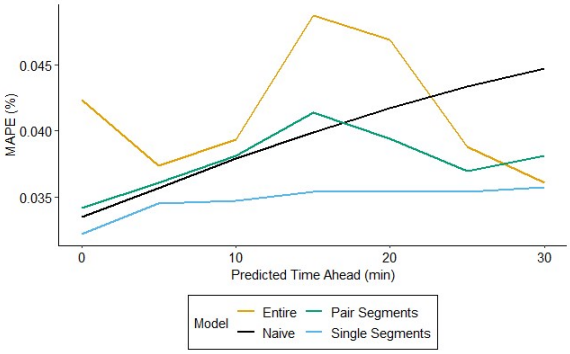
b) SH 130 SB real-time MAPE (%) for typical conditions.



c) SH 130 SB real-time MAPE (%) for atypical conditions.

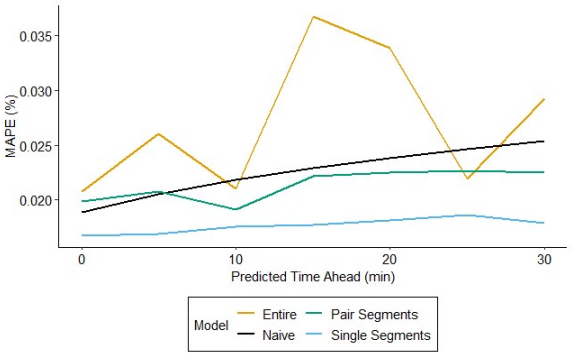


d) SH 130 SB real-time MAPE (%) for the AM peak period.

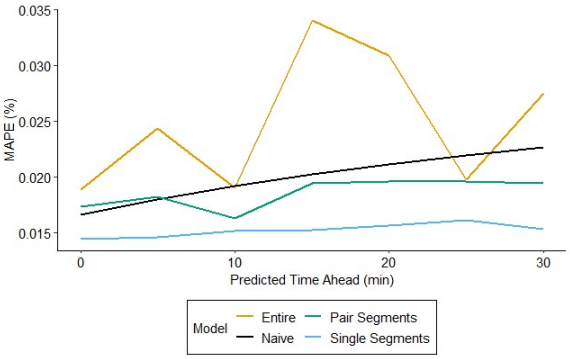


e) SH 130 SB real-time MAPE (%) for the PM peak period.

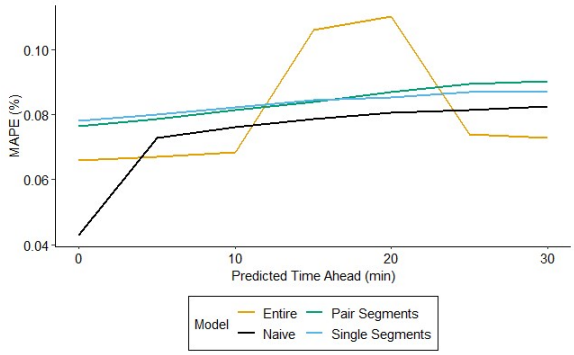
Figure 122: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 SB.



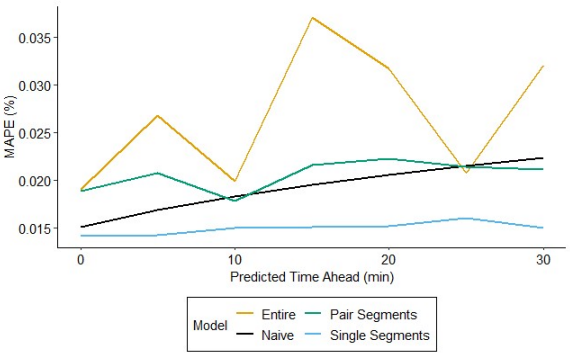
a) SH 130 NB average real-time MAPE (%).



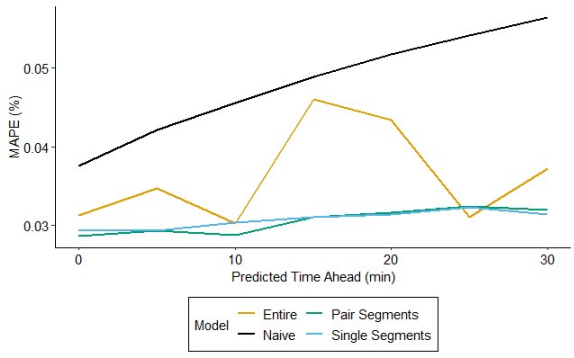
b) SH 130 NB real-time MAPE (%) for typical conditions.



c) SH 130 NB real-time MAPE (%) for atypical conditions.

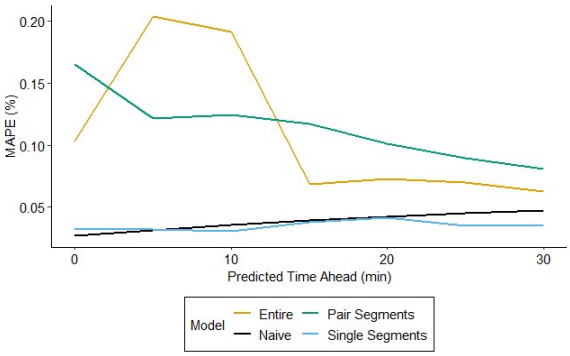


d) SH 130 NB real-time MAPE (%) for the AM peak period.

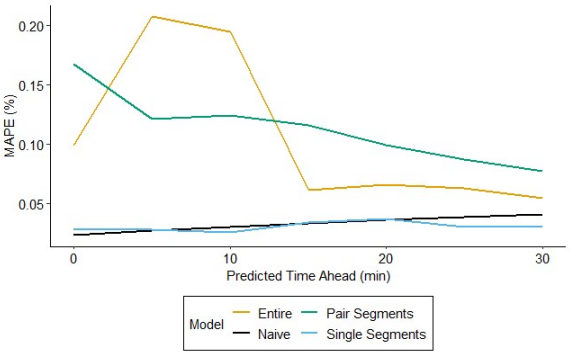


e) SH 130 NB real-time MAPE (%) for the PM peak period.

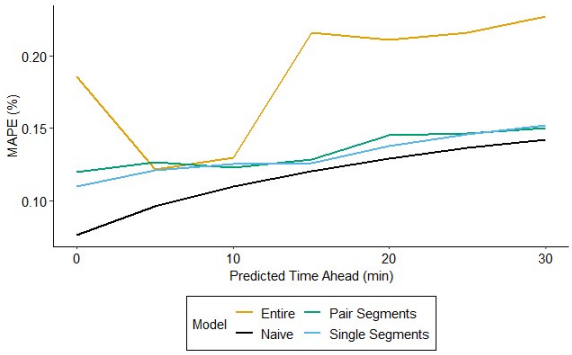
Figure 123: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 NB.



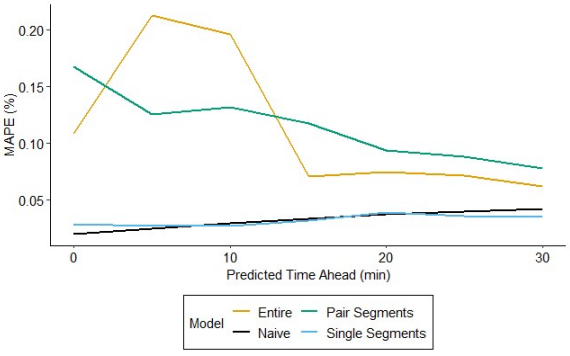
a) I-10 EB average real-time MAPE (%).



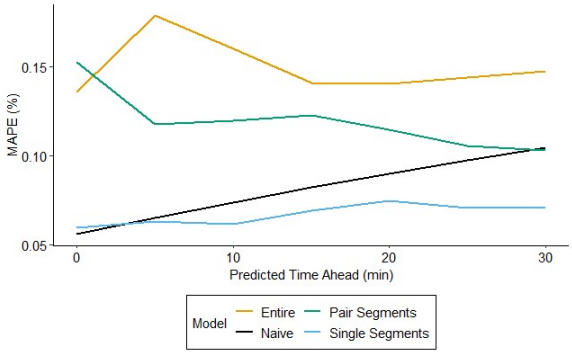
b) I-10 EB real-time MAPE (%) for typical conditions.



c) I-10 EB real-time MAPE (%) for atypical conditions.

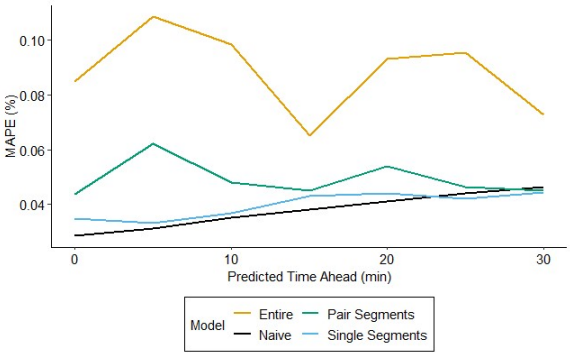


d) I-10 EB real-time MAPE (%) for the AM peak period.

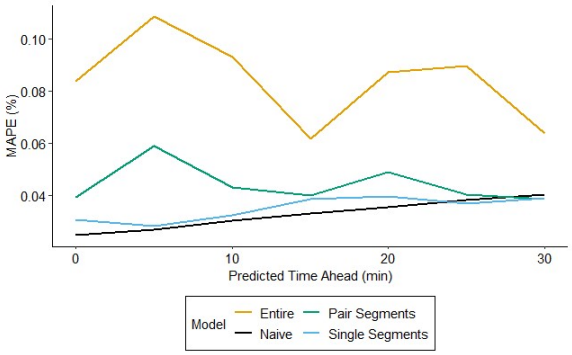


e) I-10 EB real-time MAPE (%) for the PM peak period.

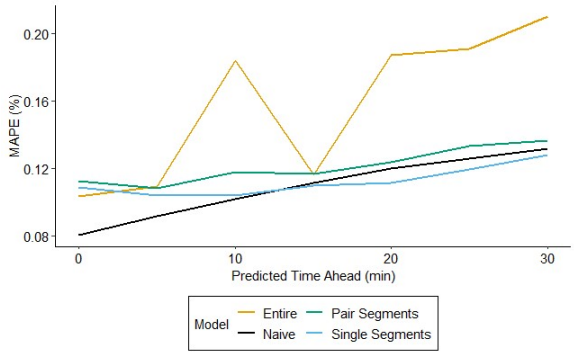
Figure 124: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 EB.



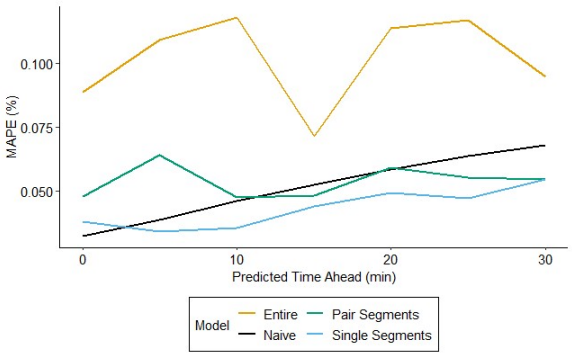
a) I-10 WB average real-time MAPE (%).



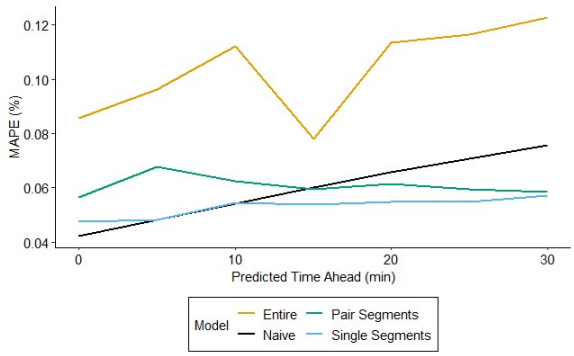
b) I-10 WB real-time MAPE (%) for typical conditions.



c) I-10 WB real-time MAPE (%) for atypical conditions.

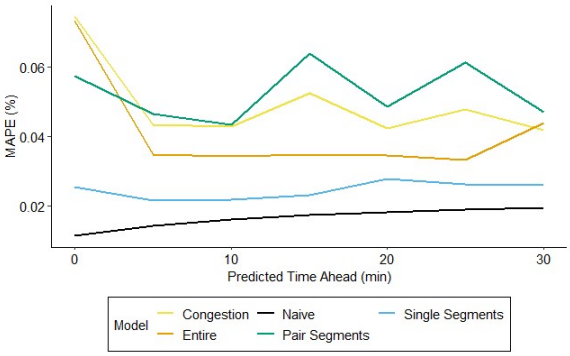


d) I-10 WB real-time MAPE (%) for the AM peak period.

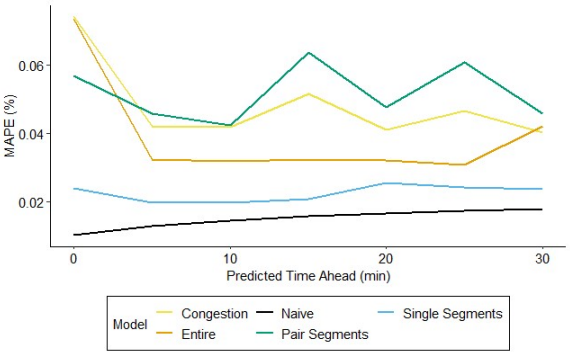


e) I-10 WB real-time MAPE (%) for the PM peak period.

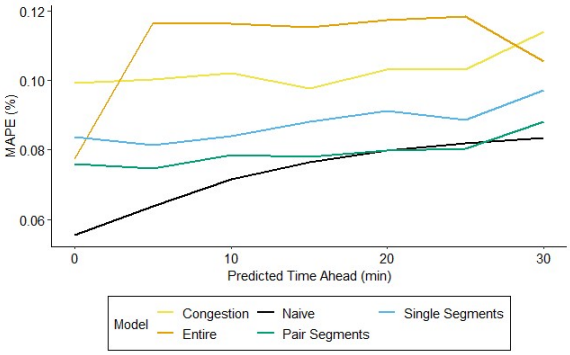
Figure 125: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 WB.



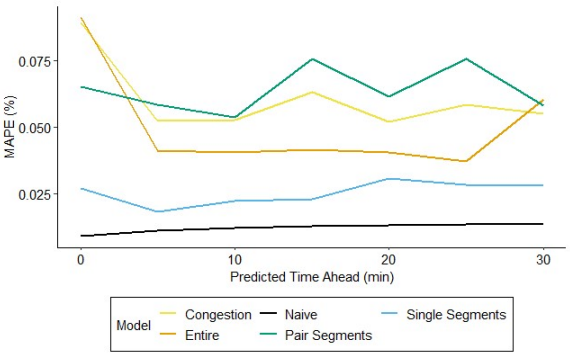
a) SL 375 EB average real-time MAPE (%).



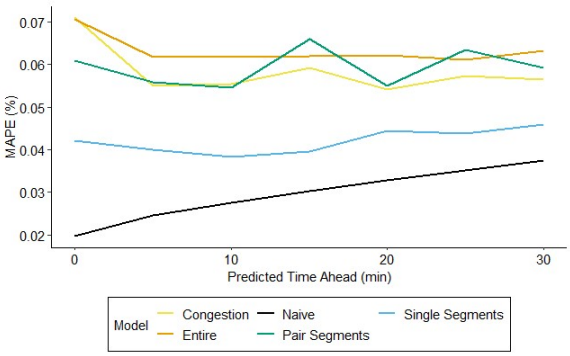
b) SL 375 EB real-time MAPE (%) for typical conditions.



c) SL 375 EB real-time MAPE (%) for atypical conditions.

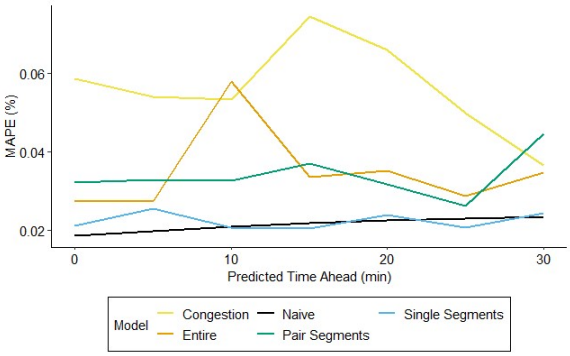


d) SL 375 EB real-time MAPE (%) for the AM peak period.

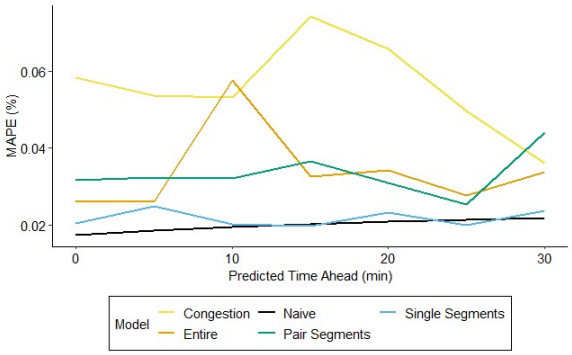


e) SL 375 EB real-time MAPE (%) for the PM peak period.

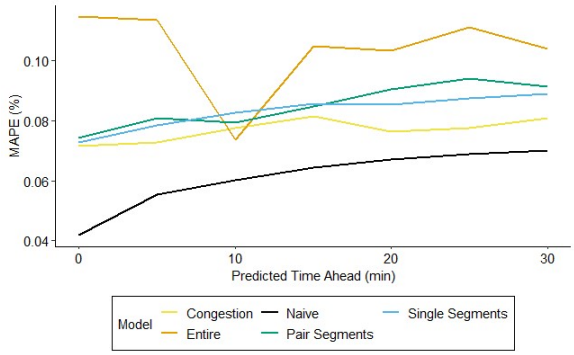
Figure 126: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 EB.



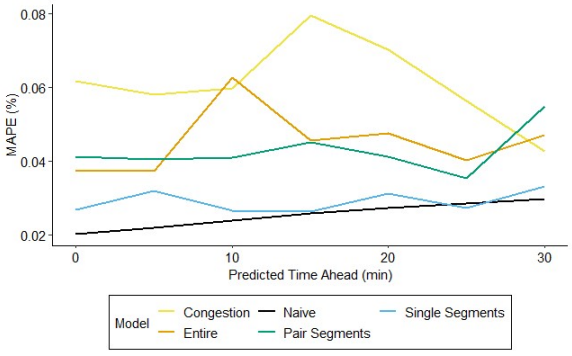
a) SL 375 WB average real-time MAPE (%).



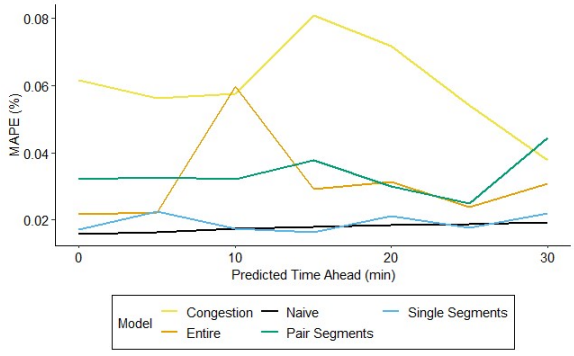
b) SL 375 WB real-time MAPE (%) for typical conditions.



c) SL 375 WB real-time MAPE (%) for atypical conditions.

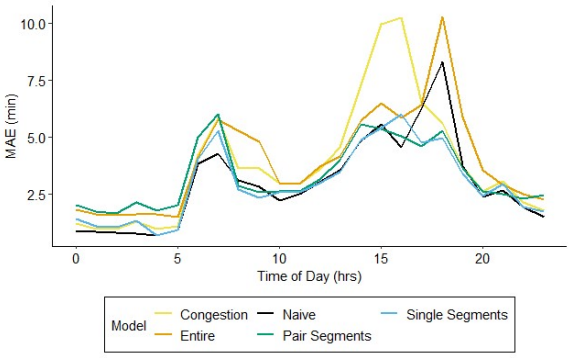


d) SL 375 WB real-time MAPE (%) for the AM peak period.

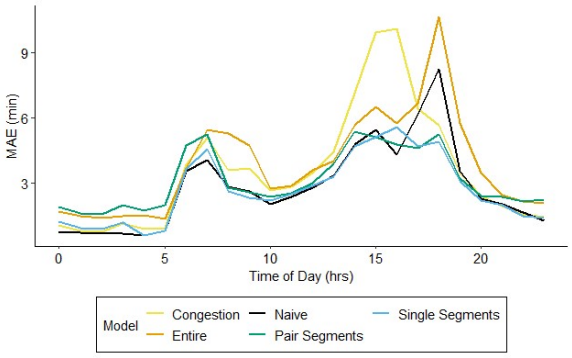


e) SL 375 WB real-time MAPE (%) for the PM peak period.

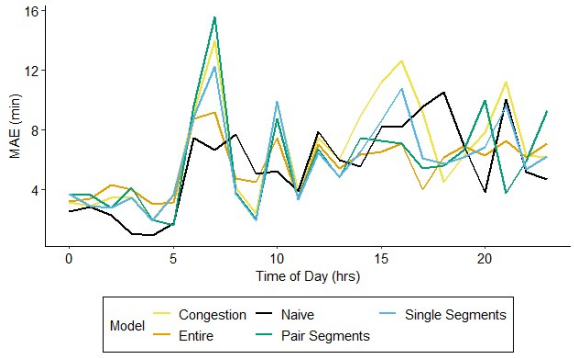
Figure 127: MAPE by prediction time ahead for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 WB.



a) I-35 SB average real-time MAE (min).

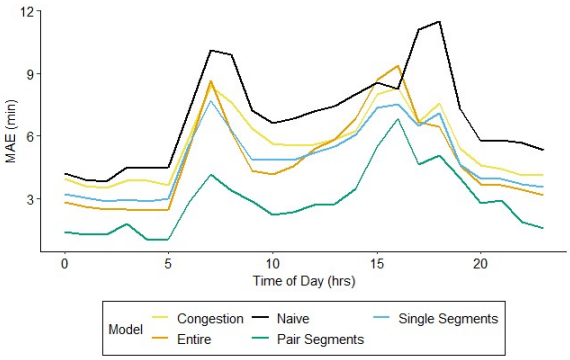


b) I-35 SB real-time MAE (min) for typical conditions.

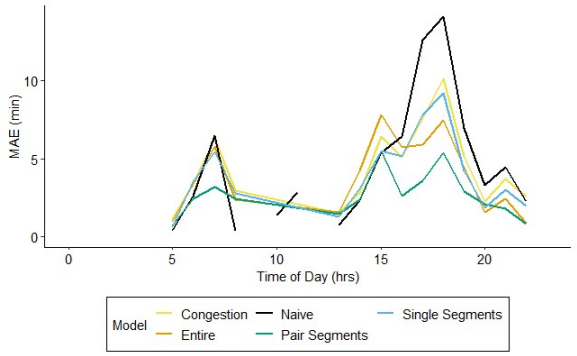


c) I-35 SB real-time MAE (min) for atypical conditions.

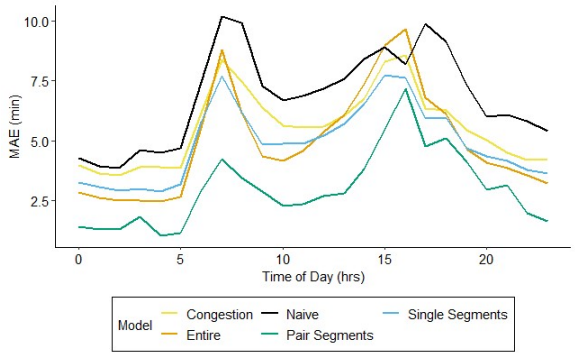
Figure 128: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 SB.



a) I-35 NB average real-time MAE (min).

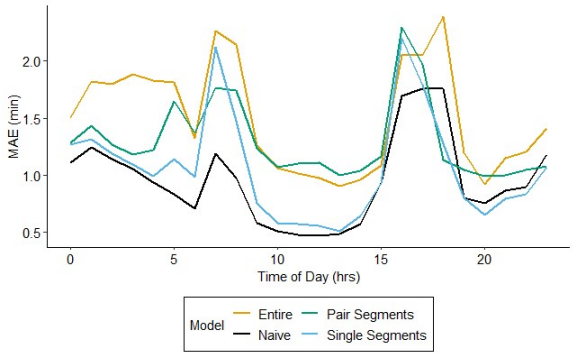


b) I-35 NB real-time MAE (min) for typical conditions.

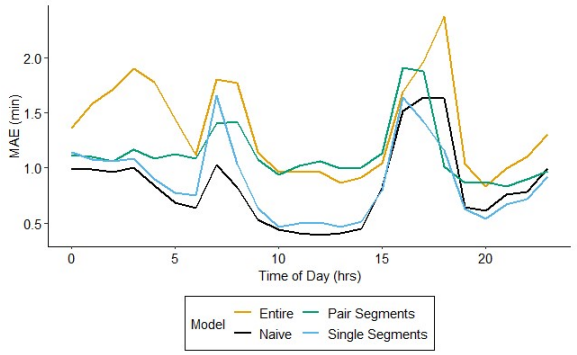


c) I-35 NB real-time MAE (min) for atypical conditions.

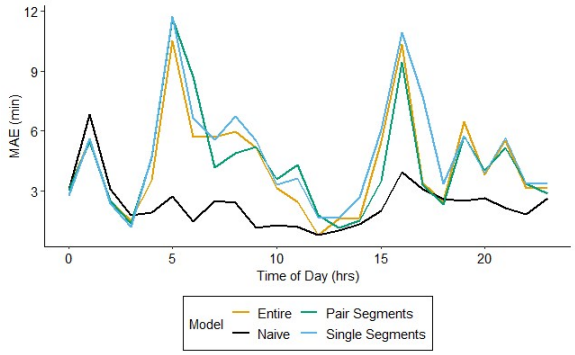
Figure 129: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 NB.



a) SH 130 SB average real-time MAE (min).

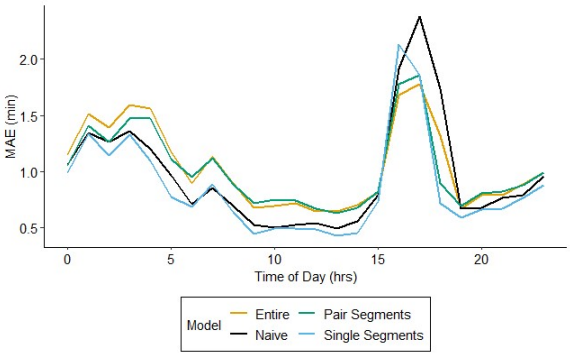


b) SH 130 SB real-time MAE (min) for typical conditions.

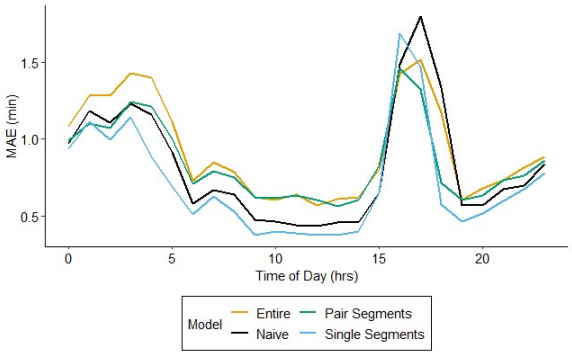


c) SH 130 SB real-time MAE (min) for atypical conditions.

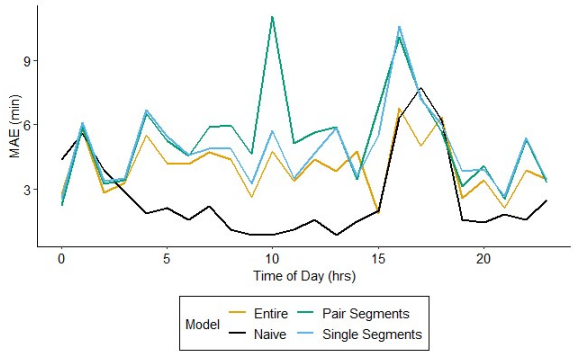
Figure 130: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 SB.



a) SH 130 NB average real-time MAE (min).

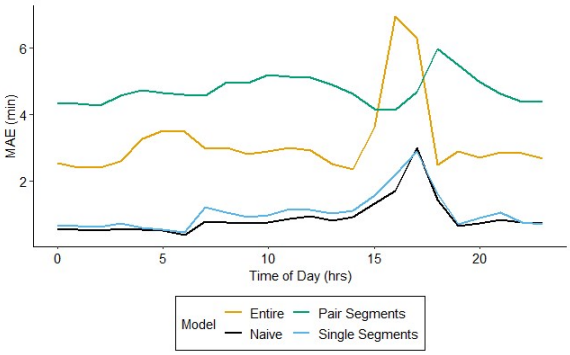


b) SH 130 NB real-time MAE (min) for typical conditions.

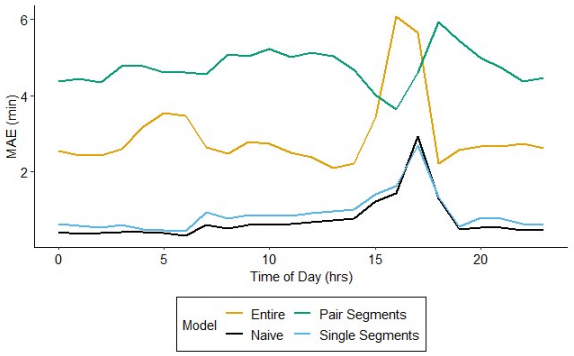


c) SH 130 NB real-time MAE (min) for atypical conditions.

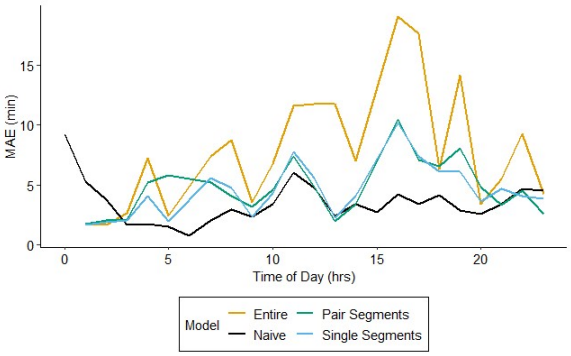
Figure 131: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 NB.



a) I-10 EB average real-time MAE (min).

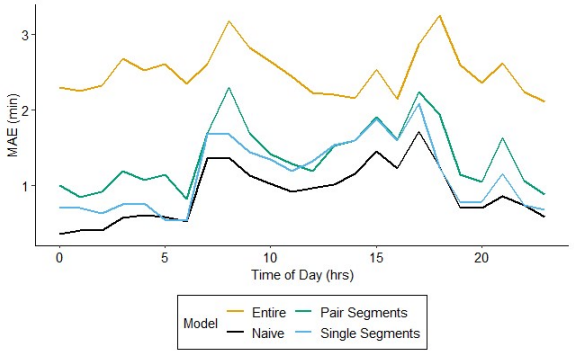


b) I-10 EB real-time MAE (min) for typical conditions.

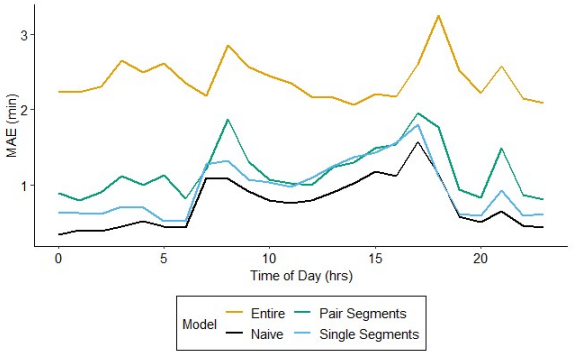


c) I-10 EB real-time MAE (min) for atypical conditions.

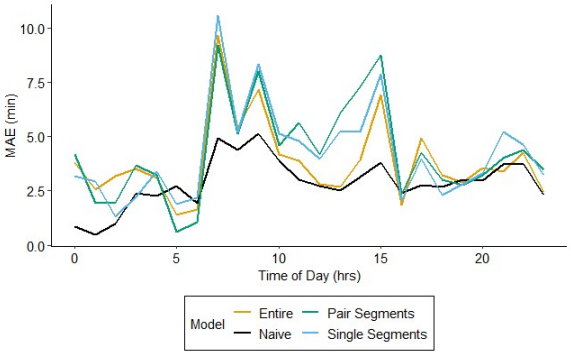
Figure 132: MAE time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 EB.



a) I-10 WB average real-time MAE (min).

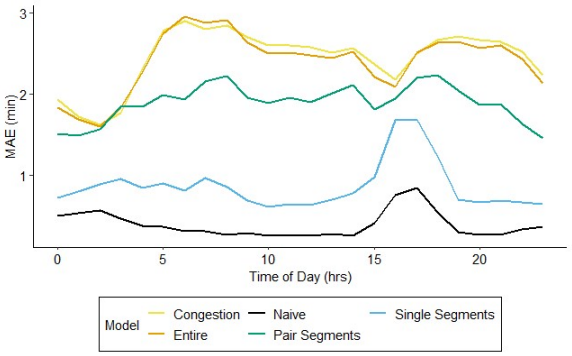


b) I-10 WB real-time MAE (min) for typical conditions.

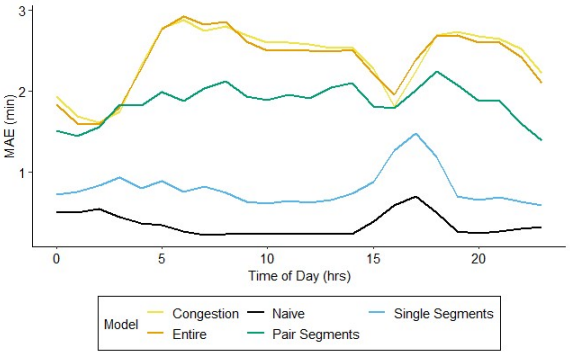


c) I-10 WB real-time MAE (min) for atypical conditions.

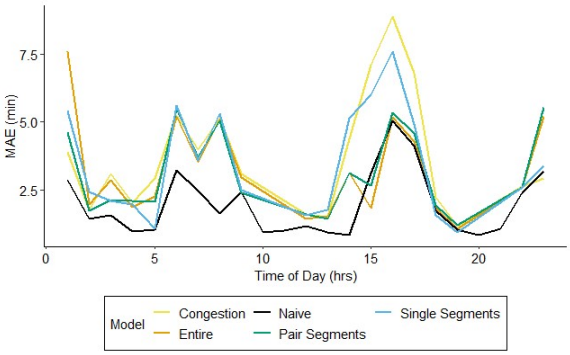
Figure 133: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 WB.



a) SL 375 EB average real-time MAE (min).

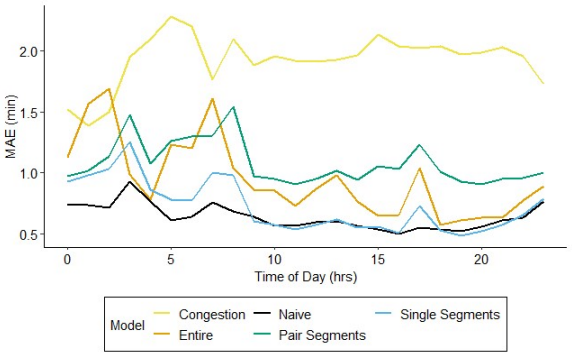


b) SL 375 EB real-time MAE (min) for typical conditions.

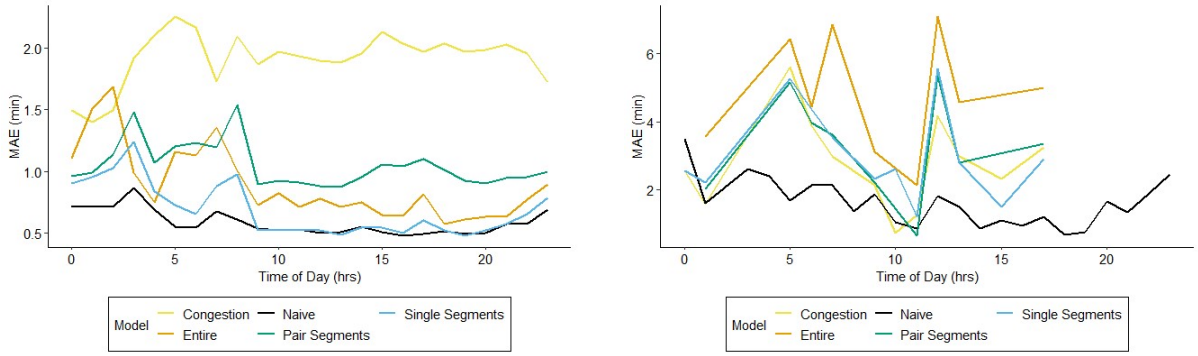


c) SL 375 EB real-time MAE (min) for atypical conditions.

Figure 134: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 EB.



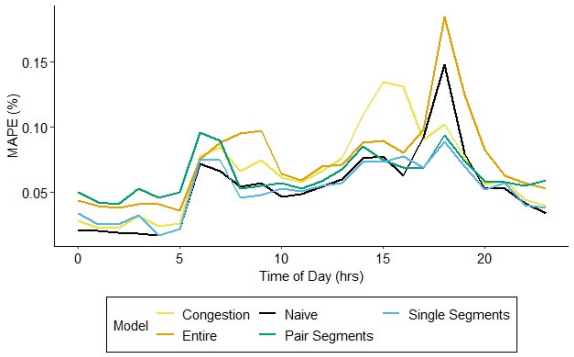
a) SL 375 WB average real-time MAE (min).



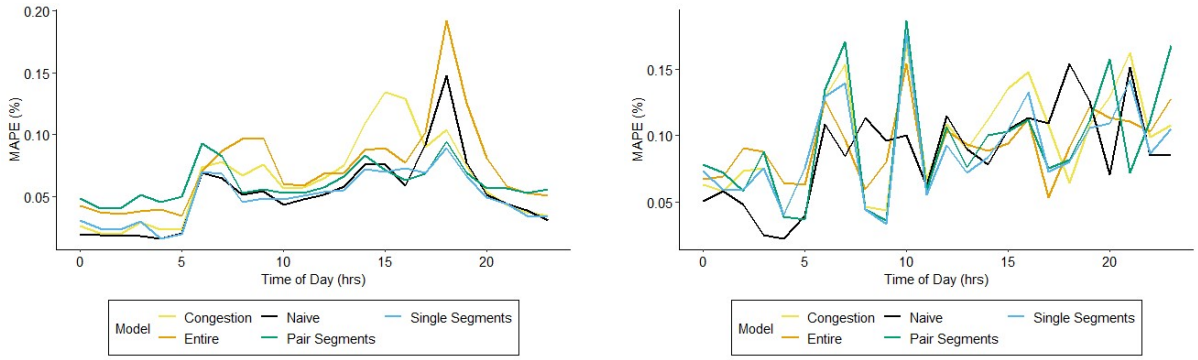
b) SL 375 WB real-time MAE (min) for typical conditions.

c) SL 375 WB real-time MAE (min) for atypical conditions.

Figure 135: MAE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 WB.



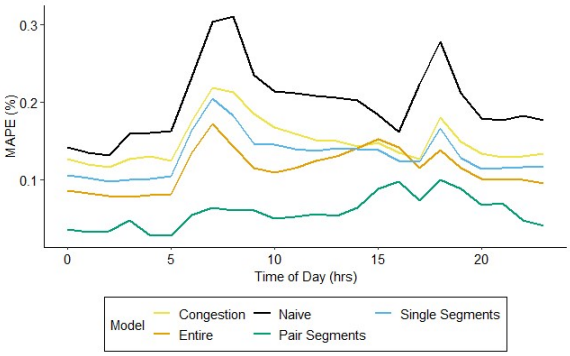
a) I-35 SB average real-time MAPE (%).



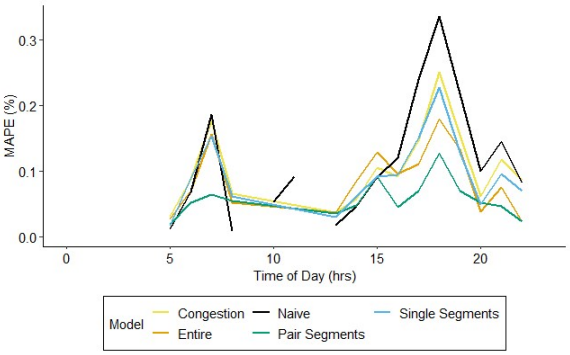
b) I-35 SB real-time MAPE (%) for typical conditions.

c) I-35 SB real-time MAPE (%) for atypical conditions.

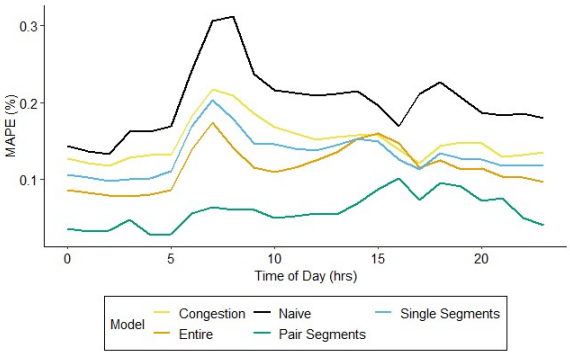
Figure 136: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 SB.



a) I-35 NB average real-time MAPE (%).

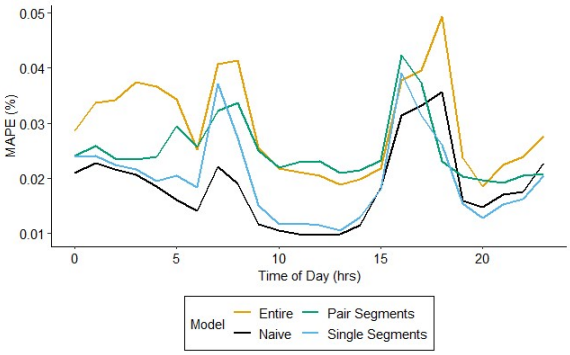


b) I-35 NB real-time MAPE (%) for typical conditions.

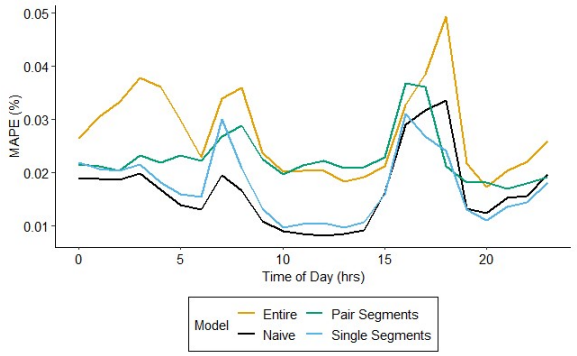


c) I-35 NB real-time MAPE (%) for atypical conditions.

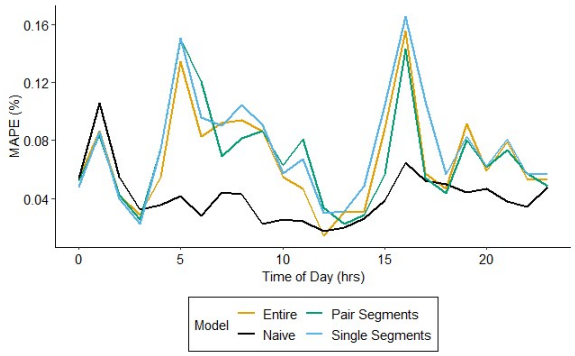
Figure 137: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-35 NB.



a) SH 130 SB average real-time MAPE (%).

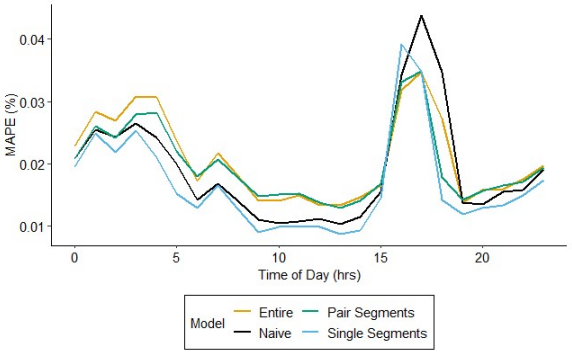


b) SH 130 SB real-time MAPE (%) for typical conditions.

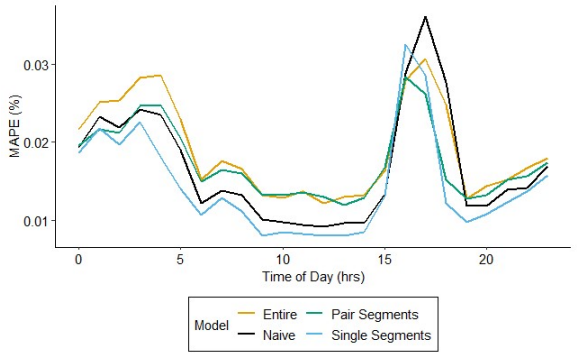


c) SH 130 SB real-time MAPE (%) for atypical conditions.

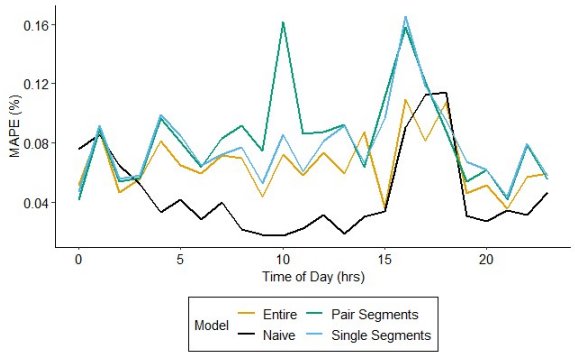
Figure 138: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 SB.



a) SH 130 NB average real-time MAPE (%).

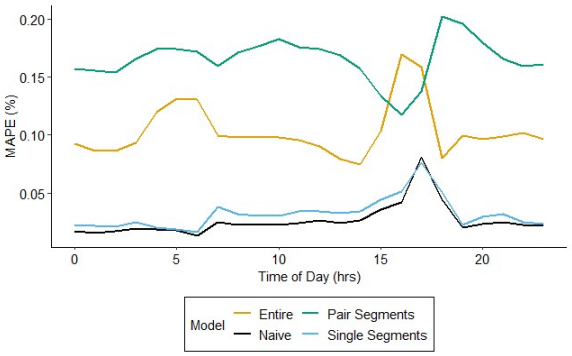


b) SH 130 NB real-time MAPE (%) for typical conditions.

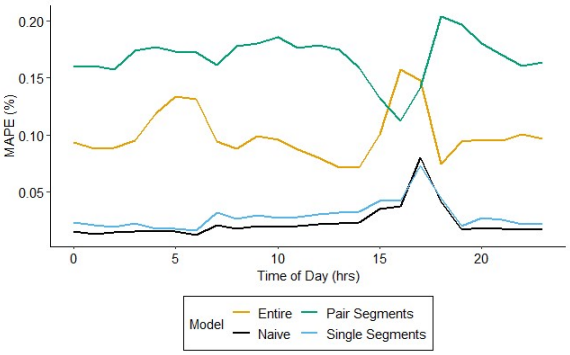


c) SH 130 NB real-time MAPE (%) for atypical conditions.

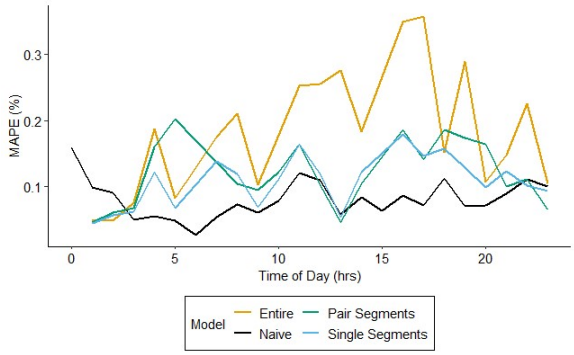
Figure 139: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SH 130 NB.



a) I-10 EB average real-time MAPE (%).

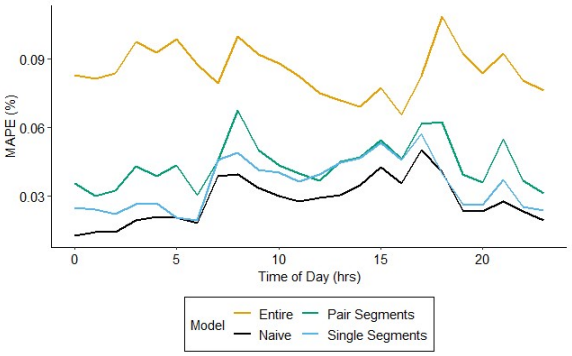


b) I-10 EB real-time MAPE (%) for typical conditions.

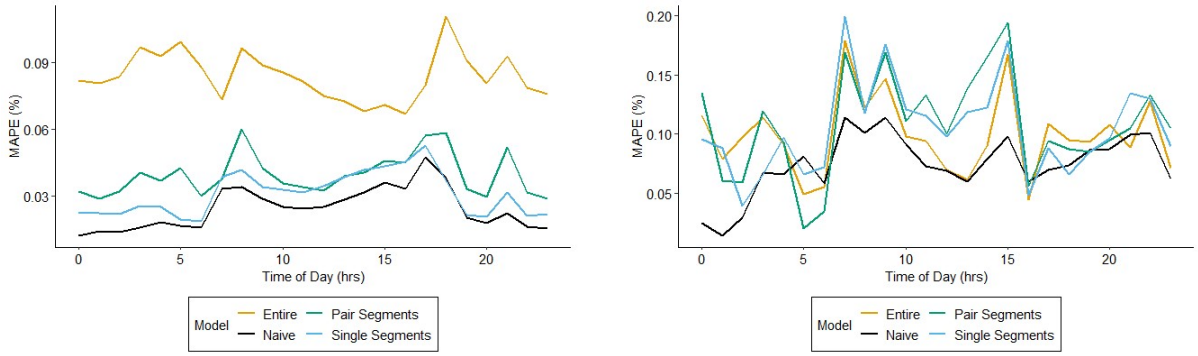


c) I-10 EB real-time MAPE (%) for atypical conditions.

Figure 140: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 EB.



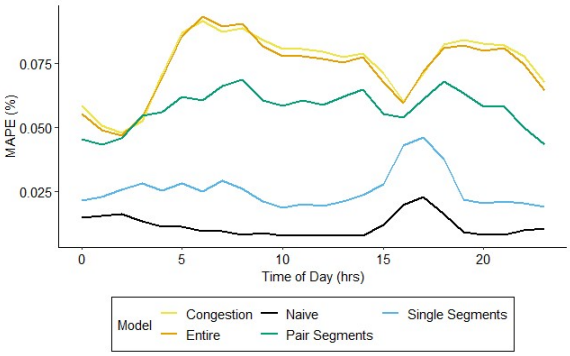
a) I-10 WB average real-time MAPE (%).



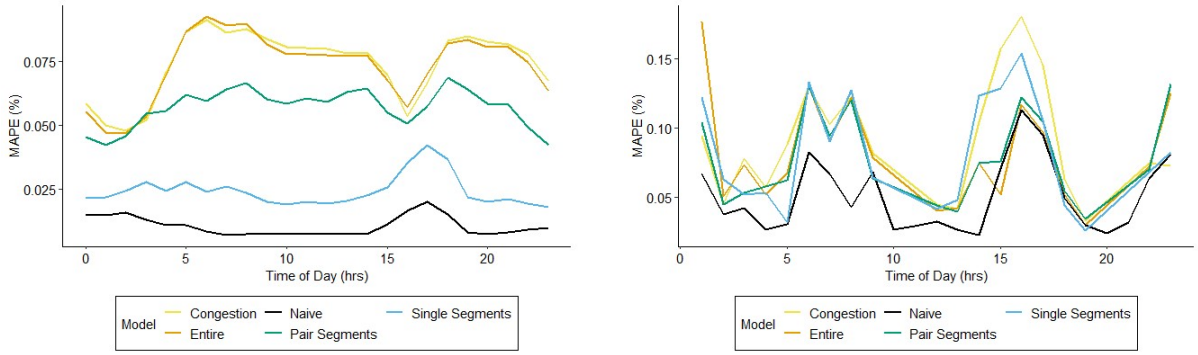
b) I-10 WB real-time MAPE (%) for typical conditions.

c) I-10 WB real-time MAPE (%) for atypical conditions.

Figure 141: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on I-10 WB.



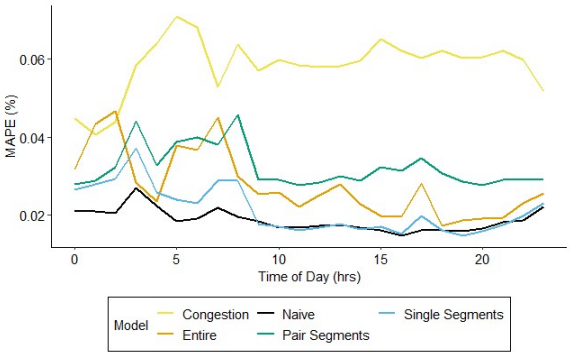
a) SL 375 EB average real-time MAPE (%).



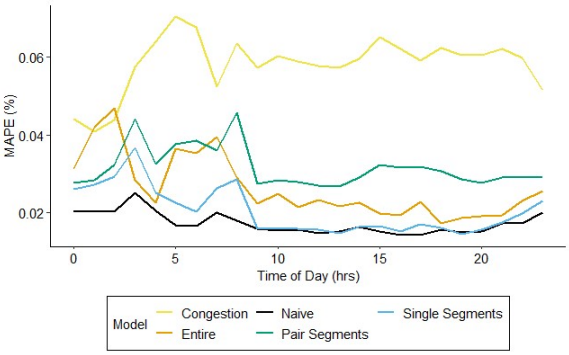
b) SL 375 EB real-time MAPE (%) for typical conditions.

c) SL 375 EB real-time MAPE (%) for atypical conditions.

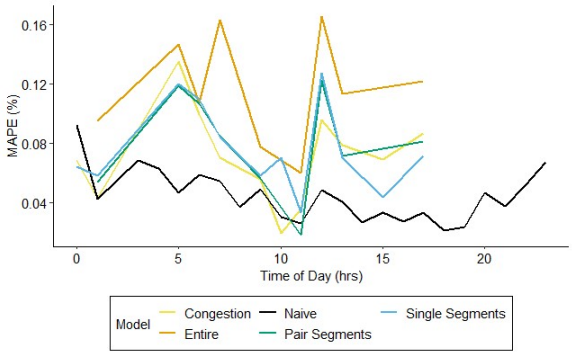
Figure 142: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 EB.



a) SL 375 WB average real-time MAPE (%).



b) SL 375 WB real-time MAPE (%) for typical conditions.



c) SL 375 WB real-time MAPE (%) for atypical conditions.

Figure 143: MAPE by time of day for naive and ML predictions using real-time data from January 2 - May 31, 2023 on SL 375 WB.

Appendix D. Incorrect Shortest Route Predictions (Atypical)

Table 26: Comparison of percentage of time incorrect predictions of shortest route are made using naive method versus ML models (atypically high travel times).

Model	Timestep	Atypically High											
		EB			WB			NB			SB		
		I-10	SL 375	**ELP**	I-10	SL 375	**ELP**	I-35	SH 130	**AUS**	I-35	SH 130	**AUS**
Naïve	0	7.77	3.74	**11.51**	8.42	3.48	**11.90**	5.58	3.52	**9.10**	24.00	1.34	**25.33**
	5	10.17	4.19	**14.36**	9.65	3.79	**13.44**	6.31	3.73	**10.04**	24.19	1.47	**25.66**
	10	12.42	4.11	**16.53**	11.41	4.10	**15.51**	7.17	3.81	**10.98**	24.40	1.57	**25.97**
	15	15.26	4.79	**20.04**	13.06	4.45	**17.51**	8.37	4.11	**12.48**	24.61	1.70	**26.31**
	20	17.65	4.94	**22.59**	14.70	4.62	**19.32**	9.53	4.19	**13.71**	25.14	1.83	**26.97**
	25	19.45	5.09	**24.53**	16.13	4.70	**20.82**	10.72	4.42	**15.14**	25.53	1.90	**27.44**
	30	21.33	5.24	**26.57**	17.58	5.02	**22.60**	11.47	4.49	**15.96**	26.02	2.07	**28.09**
Route-Level	0	67.04	0.00	**67.04**	7.47	12.12	**19.59**	10.47	1.29	**11.76**	29.04	0.96	**30.00**
	5	2.22	27.58	**29.80**	6.26	13.83	**20.09**	4.76	2.38	**7.14**	18.65	2.47	**21.12**
	10	2.22	23.27	**25.49**	70.03	0.00	**70.03**	7.21	1.97	**9.18**	24.04	1.52	**25.56**
	15	67.58	0.00	**67.58**	16.03	8.56	**24.59**	6.46	13.53	**19.99**	17.87	9.61	**27.47**
	20	67.58	0.00	**67.58**	70.03	0.00	**70.03**	7.89	12.30	**20.19**	21.63	7.64	**29.27**
	25	67.58	0.00	**67.58**	70.03	0.00	**70.03**	9.45	2.72	**12.17**	27.02	1.63	**28.65**
	30	67.58	0.00	**67.58**	70.03	0.00	**70.03**	30.80	0.48	**31.27**	46.24	0.17	**46.40**
Single Segments	0	22.49	6.88	**29.37**	16.31	1.69	**18.01**	4.34	2.02	**6.37**	23.99	1.04	**25.03**
	5	23.43	6.29	**29.72**	14.03	1.82	**15.84**	4.76	2.02	**6.78**	24.86	0.96	**25.82**
	10	24.21	5.82	**30.03**	11.43	2.86	**14.29**	5.18	2.26	**7.44**	25.12	0.83	**25.95**
	15	24.06	5.50	**29.56**	13.12	2.47	**15.58**	4.71	1.79	**6.49**	22.23	1.09	**23.32**
	20	26.26	3.93	**30.19**	15.71	2.21	**17.92**	4.83	1.71	**6.55**	23.26	1.16	**24.42**
	25	35.69	2.36	**38.05**	18.70	1.43	**20.13**	6.34	2.38	**8.71**	26.35	2.18	**28.53**
	30	33.02	2.20	**35.22**	23.25	2.60	**25.84**	5.80	3.77	**9.57**	26.78	2.45	**29.23**
Pair Segments	0	10.51	13.92	**24.43**	13.17	3.62	**16.78**	4.01	2.09	**6.10**	12.48	1.61	**14.09**
	5	23.02	5.18	**28.20**	10.65	3.11	**13.76**	3.49	2.70	**6.19**	11.49	1.45	**12.94**
	10	19.07	8.99	**28.07**	14.10	4.99	**19.09**	4.10	2.18	**6.28**	13.09	1.53	**14.62**
	15	30.65	5.99	**36.65**	15.87	4.33	**20.20**	3.92	2.09	**6.02**	12.71	2.22	**14.93**
	20	38.15	2.45	**40.60**	17.65	4.44	**22.09**	3.78	4.05	**7.83**	9.93	2.85	**12.78**
	25	42.78	0.27	**43.05**	19.87	4.88	**24.75**	4.71	3.41	**8.12**	10.49	2.62	**13.11**
	30	32.97	2.86	**35.83**	33.74	2.89	**36.63**	6.12	2.99	**9.11**	13.80	2.97	**16.77**

Table 27: Comparison of percentage of time incorrect predictions of shortest route are made using naive method versus ML models (atypically low travel times).

Model	Timestep	Atypically Low											
		EB			WB			NB			SB		
		I-10	SL 375	**ELP**	I-10	SL 375	**ELP**	I-35	SH 130	**AUS**	I-35	SH 130	**AUS**
Naïve	0	0.00	0.00	**0.00**	10.00	4.12	**14.12**	7.97	5.19	**13.16**	15.28	10.96	**26.25**
	5	0.00	0.00	**0.00**	13.33	6.67	**20.00**	3.15	3.94	**7.09**	5.26	0.00	**5.26**
	10	0.00	0.00	**0.00**	13.33	4.44	**17.78**	3.17	3.97	**7.14**	5.26	0.00	**5.26**
	15	0.00	0.00	**0.00**	11.11	4.44	**15.56**	4.00	4.00	**8.00**	5.26	0.75	**6.02**
	20	0.00	0.00	**0.00**	8.89	6.67	**15.56**	4.03	4.84	**8.87**	5.26	1.50	**6.77**
	25	0.00	0.00	**0.00**	8.89	11.11	**20.00**	4.03	5.65	**9.68**	4.48	2.24	**6.72**
	30	0.00	0.00	**0.00**	8.89	11.11	**20.00**	4.00	6.40	**10.40**	4.44	2.96	**7.41**
Route-Level	0	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	5	0.00	66.67	**66.67**	-	-	**-**	0.00	0.00	**0.00**	0.00	2.35	**2.35**
	10	0.00	50.00	**50.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	15	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	20	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	25	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	30	0.00	0.00	**0.00**	-	-	**-**	0.00	38.89	**38.89**	0.00	76.47	**76.47**
Single Segments	0	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	5	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	10	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	15	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	20	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	25	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	2.35	**2.35**
	30	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	4.71	**4.71**
Pair Segments	0	0.00	83.33	**83.33**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	5	0.00	50.00	**50.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	10	0.00	33.33	**33.33**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	15	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	20	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	25	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
	30	0.00	0.00	**0.00**	-	-	**-**	0.00	0.00	**0.00**	0.00	0.00	**0.00**

Appendix E. Real-Time Model Performance

Table 28: MAE for travel time predictions models run in real time.

Route	Model	Step	Average	AM	PM	Typical	Atypical
I-10 EB	Naive	0	0.89	0.61	2.04	0.74	3.14
I-10 EB	Naive	1	1.04	0.79	2.35	0.86	3.95
I-10 EB	Naive	2	1.18	0.94	2.64	0.97	4.47
I-10 EB	Naive	3	1.29	1.07	2.91	1.07	4.88
I-10 EB	Naive	4	1.39	1.18	3.16	1.15	5.24
I-10 EB	Naive	5	1.49	1.27	3.40	1.23	5.54
I-10 EB	Naive	6	1.57	1.34	3.63	1.30	5.79
I-10 WB	Naive	0	0.91	1.07	1.39	0.76	3.12
I-10 WB	Naive	1	1.01	1.29	1.60	0.83	3.58
I-10 WB	Naive	2	1.13	1.54	1.79	0.93	3.98
I-10 WB	Naive	3	1.24	1.77	1.98	1.02	4.35
I-10 WB	Naive	4	1.33	1.97	2.16	1.10	4.69
I-10 WB	Naive	5	1.42	2.15	2.32	1.18	4.93
I-10 WB	Naive	6	1.50	2.30	2.47	1.25	5.18
I-35 NB	Naive	0	7.05	9.08	10.30	8.15	6.86
I-35 NB	Naive	1	7.18	9.18	10.82	8.69	6.96
I-35 NB	Naive	2	7.43	9.46	11.32	9.41	7.17
I-35 NB	Naive	3	7.67	9.72	11.77	10.12	7.37
I-35 NB	Naive	4	7.90	9.98	12.20	10.79	7.56
I-35 NB	Naive	5	8.12	10.22	12.59	11.41	7.75
I-35 NB	Naive	6	8.33	10.43	12.95	11.99	7.93
I-35 SB	Naive	0	3.07	3.72	6.38	2.93	5.34
I-35 SB	Naive	1	3.40	4.25	7.00	3.20	7.15
I-35 SB	Naive	2	3.71	4.77	7.50	3.50	7.75
I-35 SB	Naive	3	4.02	5.26	7.98	3.79	8.35
I-35 SB	Naive	4	4.32	5.74	8.44	4.08	8.83
I-35 SB	Naive	5	4.61	6.22	8.85	4.36	9.37
I-35 SB	Naive	6	4.89	6.67	9.22	4.63	9.91
SH 130 NB	Naive	0	0.96	0.75	2.01	0.82	2.46
SH 130 NB	Naive	1	1.05	0.85	2.26	0.89	4.44
SH 130 NB	Naive	2	1.11	0.92	2.44	0.95	4.63
SH 130 NB	Naive	3	1.17	0.98	2.61	1.00	4.78
SH 130 NB	Naive	4	1.21	1.03	2.76	1.04	4.89
SH 130 NB	Naive	5	1.25	1.08	2.88	1.08	4.97
SH 130 NB	Naive	6	1.29	1.12	3.00	1.11	5.03
SH 130 SB	Naive	0	0.95	0.96	1.73	0.83	2.12
SH 130 SB	Naive	1	1.02	1.12	1.86	0.88	3.17
SH 130 SB	Naive	2	1.09	1.29	1.98	0.94	3.44
SH 130 SB	Naive	3	1.15	1.44	2.08	0.99	3.68

Route	Model	Step	Average	AM	PM	Typical	Atypical
SH 130 SB	Naive	4	1.21	1.58	2.18	1.04	3.89
SH 130 SB	Naive	5	1.26	1.70	2.27	1.09	4.07
SH 130 SB	Naive	6	1.31	1.81	2.34	1.13	4.21
SL 375 EB	Naive	0	0.38	0.29	0.71	0.34	2.18
SL 375 EB	Naive	1	0.49	0.37	0.88	0.44	2.49
SL 375 EB	Naive	2	0.54	0.40	0.99	0.49	2.79
SL 375 EB	Naive	3	0.59	0.42	1.08	0.53	2.98
SL 375 EB	Naive	4	0.62	0.43	1.17	0.56	3.13
SL 375 EB	Naive	5	0.64	0.44	1.25	0.58	3.22
SL 375 EB	Naive	6	0.66	0.46	1.33	0.60	3.30
SL 375 WB	Naive	0	0.63	0.69	0.53	0.58	1.51
SL 375 WB	Naive	1	0.68	0.76	0.56	0.62	2.08
SL 375 WB	Naive	2	0.72	0.83	0.59	0.66	2.26
SL 375 WB	Naive	3	0.75	0.89	0.61	0.69	2.42
SL 375 WB	Naive	4	0.77	0.94	0.63	0.71	2.54
SL 375 WB	Naive	5	0.79	0.99	0.64	0.72	2.61
SL 375 WB	Naive	6	0.81	1.03	0.66	0.74	2.66
I-10 EB	Route-Level	0	3.16	3.14	5.24	2.94	8.13
I-10 EB	Route-Level	1	5.80	5.94	5.72	5.85	4.79
I-10 EB	Route-Level	2	5.46	5.46	5.26	5.47	5.33
I-10 EB	Route-Level	3	2.31	2.19	5.75	1.96	9.43
I-10 EB	Route-Level	4	2.42	2.29	5.71	2.09	9.25
I-10 EB	Route-Level	5	2.36	2.22	5.87	2.01	9.46
I-10 EB	Route-Level	6	2.19	2.00	6.04	1.81	9.90
I-10 EB	Pair	0	4.74	4.72	4.93	4.73	4.81
I-10 EB	Pair	1	3.52	3.56	4.03	3.44	5.43
I-10 EB	Pair	2	3.60	3.73	4.07	3.53	5.25
I-10 EB	Pair	3	3.41	3.35	4.22	3.32	5.55
I-10 EB	Pair	4	3.01	2.71	4.13	2.87	6.29
I-10 EB	Pair	5	2.69	2.55	3.92	2.53	6.41
I-10 EB	Pair	6	2.45	2.27	3.86	2.26	6.59
I-10 EB	Single	0	1.03	0.89	2.22	0.89	4.67
I-10 EB	Single	1	1.07	0.88	2.49	0.89	5.13
I-10 EB	Single	2	1.01	0.89	2.44	0.82	5.33
I-10 EB	Single	3	1.22	1.00	2.68	1.04	5.35
I-10 EB	Single	4	1.32	1.21	2.90	1.12	5.89
I-10 EB	Single	5	1.18	1.16	2.87	0.95	6.26
I-10 EB	Single	6	1.21	1.17	2.90	0.98	6.46

Route	Model	Step	Average	AM	PM	Typical	Atypical
I-10 WB	Route-Level	0	2.50	2.71	2.76	2.39	4.21
I-10 WB	Route-Level	1	3.16	3.29	3.13	3.08	4.40
I-10 WB	Route-Level	2	3.04	3.81	4.13	2.73	7.99
I-10 WB	Route-Level	3	1.99	2.31	2.71	1.80	4.86
I-10 WB	Route-Level	4	2.91	3.70	4.18	2.58	8.14
I-10 WB	Route-Level	5	2.98	3.81	4.29	2.64	8.29
I-10 WB	Route-Level	6	2.41	3.28	4.62	1.98	9.07
I-10 WB	Pair	0	1.38	1.60	1.92	1.17	4.62
I-10 WB	Pair	1	1.87	2.05	2.25	1.69	4.46
I-10 WB	Pair	2	1.50	1.62	2.10	1.27	4.92
I-10 WB	Pair	3	1.43	1.68	2.04	1.19	4.90
I-10 WB	Pair	4	1.68	2.02	2.11	1.44	5.21
I-10 WB	Pair	5	1.49	1.95	2.13	1.20	5.68
I-10 WB	Pair	6	1.46	1.97	2.07	1.16	5.79
I-10 WB	Single	0	1.13	1.29	1.63	0.94	4.54
I-10 WB	Single	1	1.08	1.17	1.69	0.86	4.25
I-10 WB	Single	2	1.17	1.22	1.85	0.96	4.27
I-10 WB	Single	3	1.35	1.48	1.85	1.13	4.57
I-10 WB	Single	4	1.39	1.66	1.89	1.16	4.68
I-10 WB	Single	5	1.35	1.64	1.92	1.10	5.06
I-10 WB	Single	6	1.43	1.90	2.00	1.16	5.40
I-35 NB	Route-Level	0	4.85	6.75	7.47	4.80	4.82
I-35 NB	Route-Level	1	5.41	6.79	7.41	5.47	5.37
I-35 NB	Route-Level	2	4.82	6.57	7.43	4.82	4.78
I-35 NB	Route-Level	3	5.73	7.41	7.36	5.45	5.72
I-35 NB	Route-Level	4	5.31	7.14	7.76	5.02	5.30
I-35 NB	Route-Level	5	5.20	7.32	8.03	4.86	5.19
I-35 NB	Route-Level	6	9.27	11.82	12.15	7.97	9.35
I-35 NB	Pair	0	2.91	3.48	5.51	3.10	2.90
I-35 NB	Pair	1	2.88	3.40	5.55	2.95	2.87
I-35 NB	Pair	2	3.10	3.59	5.65	3.19	3.09
I-35 NB	Pair	3	3.02	3.88	5.81	2.96	3.03
I-35 NB	Pair	4	3.00	3.71	5.70	3.03	2.99

Route	Model	Step	Average	AM	PM	Typical	Atypical
I-35 NB	Pair	5	2.31	2.57	2.81	2.76	2.27
I-35 NB	Pair	6	1.75	1.13	1.15	1.82	1.75
I-35 NB	Congestion Paths	0	5.55	7.32	7.52	5.56	5.52
I-35 NB	Congestion Paths	1	5.70	7.18	7.59	5.79	5.65
I-35 NB	Congestion Paths	2	5.32	6.91	7.42	5.31	5.28
I-35 NB	Congestion Paths	3	5.47	7.26	7.55	5.24	5.46
I-35 NB	Congestion Paths	4	5.41	7.27	7.94	5.02	5.41
I-35 NB	Congestion Paths	5	5.51	7.46	7.99	5.03	5.52
I-35 NB	Congestion Paths	6	5.85	7.76	8.06	5.19	5.87
I-35 NB	Single	0	4.89	6.53	7.03	5.09	4.84
I-35 NB	Single	1	5.06	6.59	7.03	5.18	5.02
I-35 NB	Single	2	4.92	6.53	7.12	4.91	4.89
I-35 NB	Single	3	4.95	6.51	7.17	4.79	4.93
I-35 NB	Single	4	4.85	5.81	6.93	4.72	4.82
I-35 NB	Single	5	4.02	3.81	3.80	3.65	4.03
I-35 NB	Single	6	3.30	2.16	1.59	1.92	3.41
I-35 SB	Route-Level	0	3.96	5.04	7.54	3.83	5.83
I-35 SB	Route-Level	1	4.66	5.36	7.66	4.56	6.16
I-35 SB	Route-Level	2	4.08	5.06	7.41	3.93	6.35
I-35 SB	Route-Level	3	4.09	5.03	7.47	3.94	6.15
I-35 SB	Route-Level	4	8.90	11.23	18.33	8.70	11.81
I-35 SB	Route-Level	5	4.44	5.34	7.38	4.31	6.43
I-35 SB	Route-Level	6	10.55	11.17	13.61	10.57	10.64
I-35 SB	Pair	0	3.30	4.63	4.96	3.13	6.03
I-35 SB	Pair	1	3.03	4.39	5.08	2.83	6.02
I-35 SB	Pair	2	3.32	4.80	4.60	3.14	5.99
I-35 SB	Pair	3	3.45	5.06	3.01	3.32	5.44
I-35 SB	Pair	4	2.97	4.65	1.82	2.85	4.72
I-35 SB	Pair	5	2.38	3.22	1.15	2.28	3.72
I-35 SB	Pair	6	1.71	1.78	0.71	1.64	2.91
I-35 SB	Congestion Paths	0	3.78	4.57	7.46	3.53	7.21
I-35 SB	Congestion Paths	1	3.99	4.35	8.27	3.75	7.42

Route	Model	Step	Average	AM	PM	Typical	Atypical
I-35 SB	Congestion Paths	2	4.13	4.22	8.94	3.87	7.77
I-35 SB	Congestion Paths	3	4.23	4.45	8.93	3.97	7.80
I-35 SB	Congestion Paths	4	4.41	4.73	9.16	4.14	8.16
I-35 SB	Congestion Paths	5	4.48	4.61	9.92	4.20	8.50
I-35 SB	Congestion Paths	6	4.78	3.54	10.57	4.55	7.90
I-35 SB	Single	0	2.94	3.99	5.23	2.70	6.30
I-35 SB	Single	1	3.23	4.49	5.21	3.00	6.34
I-35 SB	Single	2	3.05	4.53	4.99	2.83	6.06
I-35 SB	Single	3	3.07	4.73	3.36	2.89	5.42
I-35 SB	Single	4	2.21	4.12	1.66	2.10	3.62
I-35 SB	Single	5	1.85	2.97	1.04	1.74	3.20
I-35 SB	Single	6	1.36	1.62	0.54	1.28	2.35
SH 130 NB	Route-Level	0	1.04	0.97	1.59	0.92	4.10
SH 130 NB	Route-Level	1	1.29	1.34	1.75	1.17	4.18
SH 130 NB	Route-Level	2	1.06	1.02	1.54	0.93	4.29
SH 130 NB	Route-Level	3	1.83	1.85	2.32	1.64	6.65
SH 130 NB	Route-Level	4	1.70	1.60	2.20	1.49	6.90
SH 130 NB	Route-Level	5	1.10	1.06	1.58	0.96	4.67
SH 130 NB	Route-Level	6	1.45	1.60	1.87	1.32	4.60
SH 130 NB	Pair	0	1.02	0.98	1.51	0.84	4.87
SH 130 NB	Pair	1	1.06	1.07	1.54	0.88	5.03
SH 130 NB	Pair	2	0.98	0.94	1.52	0.79	5.20
SH 130 NB	Pair	3	1.13	1.12	1.62	0.94	5.37
SH 130 NB	Pair	4	1.15	1.15	1.65	0.95	5.56
SH 130 NB	Pair	5	1.16	1.11	1.69	0.95	5.71
SH 130 NB	Pair	6	1.15	1.10	1.67	0.94	5.76
SH 130 NB	Single	0	0.86	0.74	1.56	0.71	4.90
SH 130 NB	Single	1	0.87	0.75	1.56	0.72	5.02
SH 130 NB	Single	2	0.90	0.78	1.61	0.74	5.17
SH 130 NB	Single	3	0.91	0.78	1.66	0.75	5.31
SH 130 NB	Single	4	0.93	0.79	1.67	0.77	5.36
SH 130 NB	Single	5	0.96	0.83	1.72	0.79	5.47
SH 130 NB	Single	6	0.92	0.78	1.68	0.75	5.47
SH 130 SB	Route-Level	0	1.50	1.91	2.16	1.35	4.15

Route	Model	Step	Average	AM	PM	Typical	Atypical
SH 130 SB	Route-Level	1	1.29	1.74	1.94	1.11	4.49
SH 130 SB	Route-Level	2	1.44	1.94	2.02	1.28	4.44
SH 130 SB	Route-Level	3	2.11	2.77	2.50	1.86	6.81
SH 130 SB	Route-Level	4	2.01	2.68	2.43	1.74	7.00
SH 130 SB	Route-Level	5	1.56	2.19	2.00	1.38	4.86
SH 130 SB	Route-Level	6	1.27	1.82	1.89	1.07	4.85
SH 130 SB	Pair	0	1.30	1.62	1.80	1.12	4.08
SH 130 SB	Pair	1	1.57	1.98	1.88	1.40	4.23
SH 130 SB	Pair	2	1.72	2.14	1.98	1.55	4.46
SH 130 SB	Pair	3	1.90	2.39	2.14	1.73	4.70
SH 130 SB	Pair	4	1.75	2.36	2.06	1.55	5.07
SH 130 SB	Pair	5	1.53	2.21	1.96	1.29	5.44
SH 130 SB	Pair	6	1.56	2.29	2.03	1.30	5.70
SH 130 SB	Single	0	1.06	1.52	1.75	0.87	4.81
SH 130 SB	Single	1	1.11	1.55	1.89	0.91	5.05
SH 130 SB	Single	2	1.10	1.61	1.91	0.88	5.36
SH 130 SB	Single	3	1.11	1.72	1.96	0.88	5.59
SH 130 SB	Single	4	1.13	1.76	1.96	0.89	5.72
SH 130 SB	Single	5	1.16	1.81	1.96	0.91	5.87
SH 130 SB	Single	6	1.15	1.84	1.98	0.90	5.92
SL 375 EB	Route-Level	0	2.40	2.91	2.41	2.38	3.06
SL 375 EB	Route-Level	1	1.20	1.35	2.32	1.09	4.74
SL 375 EB	Route-Level	2	1.18	1.33	2.32	1.08	4.74
SL 375 EB	Route-Level	3	1.20	1.36	2.32	1.09	4.70
SL 375 EB	Route-Level	4	1.19	1.33	2.33	1.08	4.78
SL 375 EB	Route-Level	5	1.15	1.23	2.30	1.04	4.81
SL 375 EB	Route-Level	6	1.48	1.95	2.32	1.39	4.30
SL 375 EB	Pair	0	1.89	2.10	2.12	1.86	3.02
SL 375 EB	Pair	1	1.54	1.89	1.95	1.50	3.00
SL 375 EB	Pair	2	1.44	1.73	1.90	1.38	3.15
SL 375 EB	Pair	3	2.10	2.43	2.26	2.07	3.10
SL 375 EB	Pair	4	1.60	1.98	1.92	1.56	3.22
SL 375 EB	Pair	5	2.01	2.43	2.19	1.98	3.21
SL 375 EB	Pair	6	1.56	1.88	2.12	1.50	3.58

Route	Model	Step	Average	AM	PM	Typical	Atypical
SL 375 EB	Congestion Paths	0	2.44	2.84	2.45	2.41	4.19
SL 375 EB	Congestion Paths	1	1.43	1.68	1.97	1.38	4.31
SL 375 EB	Congestion Paths	2	1.42	1.69	1.99	1.36	4.38
SL 375 EB	Congestion Paths	3	1.73	2.02	2.07	1.68	4.18
SL 375 EB	Congestion Paths	4	1.40	1.67	1.94	1.34	4.42
SL 375 EB	Congestion Paths	5	1.57	1.87	2.04	1.52	4.42
SL 375 EB	Congestion Paths	6	1.39	1.76	2.05	1.32	4.92
SL 375 EB	Single	0	0.86	0.87	1.53	0.79	3.59
SL 375 EB	Single	1	0.75	0.63	1.47	0.67	3.29
SL 375 EB	Single	2	0.75	0.76	1.40	0.66	3.40
SL 375 EB	Single	3	0.79	0.77	1.45	0.70	3.58
SL 375 EB	Single	4	0.94	1.02	1.64	0.85	3.72
SL 375 EB	Single	5	0.89	0.94	1.61	0.81	3.61
SL 375 EB	Single	6	0.90	0.94	1.71	0.80	3.96
SL 375 WB	Route-Level	0	0.95	1.28	0.76	0.89	4.69
SL 375 WB	Route-Level	1	0.94	1.30	0.78	0.88	4.64
SL 375 WB	Route-Level	2	1.91	2.07	2.00	1.90	2.95
SL 375 WB	Route-Level	3	1.13	1.55	1.00	1.08	4.29
SL 375 WB	Route-Level	4	1.18	1.61	1.07	1.14	4.23
SL 375 WB	Route-Level	5	0.98	1.38	0.83	0.93	4.54
SL 375 WB	Route-Level	6	1.17	1.60	1.05	1.13	4.25
SL 375 WB	Pair	0	1.08	1.38	1.09	1.05	3.01
SL 375 WB	Pair	1	1.10	1.36	1.11	1.07	3.31
SL 375 WB	Pair	2	1.09	1.38	1.10	1.06	3.24
SL 375 WB	Pair	3	1.24	1.52	1.28	1.21	3.46
SL 375 WB	Pair	4	1.07	1.39	1.02	1.03	3.72
SL 375 WB	Pair	5	0.89	1.21	0.86	0.85	3.87
SL 375 WB	Pair	6	1.49	1.83	1.50	1.46	3.73
SL 375 WB	Congestion Paths	0	1.93	2.02	2.03	1.92	2.77
SL 375 WB	Congestion Paths	1	1.78	1.91	1.86	1.77	2.84
SL 375 WB	Congestion Paths	2	1.76	1.96	1.91	1.74	3.05

Route	Model	Step	Average	AM	PM	Typical	Atypical
SL 375 WB	Congestion Paths	3	2.45	2.60	2.67	2.45	3.10
SL 375 WB	Congestion Paths	4	2.17	2.30	2.37	2.16	2.96
SL 375 WB	Congestion Paths	5	1.65	1.86	1.79	1.63	3.07
SL 375 WB	Congestion Paths	6	1.21	1.42	1.25	1.19	3.23
SL 375 WB	Single	0	0.72	0.92	0.59	0.69	2.93
SL 375 WB	Single	1	0.86	1.08	0.76	0.83	3.21
SL 375 WB	Single	2	0.71	0.92	0.60	0.68	3.41
SL 375 WB	Single	3	0.70	0.91	0.56	0.66	3.53
SL 375 WB	Single	4	0.81	1.06	0.72	0.78	3.52
SL 375 WB	Single	5	0.71	0.95	0.60	0.67	3.62
SL 375 WB	Single	6	0.83	1.13	0.75	0.79	3.66

Center for Transportation Research
The University of Texas at Austin
3925 W. Braker Lane, 4th Floor
Austin TX 78759
512.232.3100
http://ctr.utexas.edu