City of Atlanta Research Project

Final Report

SMART CITIES ATLANTA - NORTH AVENUE

By
Michael Hunter, Ph.D., Principal Investigator
Randall Guensler, Ph.D., co-Principal Investigator
Angshuman Guin, Ph.D., co-Principal Investigator
Abhilasha Saroj, Ph.D. Candidate
Somdut Roy, Ph.D. Candidate

School of Civil and Environmental Engineering, Georgia Institute of Technology

December 2019

The contents of this report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the City of Atlanta. This report does not constitute a standard, specification, or regulation.

TABLE OF CONTENTS

1.0 INTRODUCTION 1

1.1. North Avenue ITS Demonstration Testbed Overview 1

1.2. Project Background 2

2.0 LITERATURE REVIEW 5

2.1. Connected Corridor Deployments and Testbed Studies 5

2.2. Real-Time Traffic Data-Driven Simulation Modelling 6

2.3. Traffic Data Imputation Methodologies 8

3.0 PHASE 1: HYBRID MODEL ARCHITECTURE, DATA DESCRIPTION, AND PERFORMANCE TEST 9

3.1. Introduction 9

3.2. Model Architecture 10

3.2.1. Traffic Simulation Model 11

3.2.2. Injecting Real-Time Intersection Signal State and Volume Count Data 11

3.2.3. Dynamic Performance Evaluation Visualizations 12

3.3. Result and Discussion 14

3.3.1. Real-time Performance Measures Computation 14

3.3.2. Model Sensitivity to Real-Time Input 15

3.4. Conclusion and Future Work 18

4.0 UPDATED MODEL ARCHITECTURE, DATA ISSUES, AND MODEL PERFORMANCE SENSITIVITY TO VOLUME IMPUTATIONS EXPERIMENT 20

4.1. Introduction 20

4.2. Updated Real-Time Data-Driven Simulation Model Architecture 21

4.2.1. Real-Time Raw Data Stream Processing Module 22

4.2.2. Dynamic Data-Driven Traffic Simulation Module 23

4.2.3. Dynamic Performance Measures Visualization Module 25

4.2.4. Data Request Transactions Management Module – Flask Web Server 25

4.3. Software Architecture 26

4.3.1. Extract, Transform, and Load Data 26

4.3.2. Data Retrieval and Coordination via Flask 28

4.3.3. Real-time Trajectory and Signal Data-processing 30

4.3.4. Trajectory Data Condensation 31

4.3.5. KPI calculation 34

4.3.1. Visualization frontend 36

4.4. Investigation of Real-Time Data Streams 36
4.4.1. Volume Data Streams 37
4.4.2. Signal Data Streams 42
4.5. Sensitivity Analysis Experiment Methodology 45
4.5.1. Experiment Design 45
4.6. Results 58
4.6.1. Intermittent Data Loss 60
4.7. Discussion 63
4.8. Conclusion 64
5.0 CONCLUSIONS 65
5.1. Recommendations and limitations 66
5.2. Next Steps 66
6.0 REFERENCES 67
APPENDIX A: SIDEWALK, RAMP, AND CURB CUT INVENTORY AND ASSESSMENT A1

Table of Figures

Figure 1: North Avenue corridor and local neighborhoods 3
Figure 2: Households in each North Avenue corridor neighborhood 3
Figure 3: Hybrid model architecture. 10
Figure 4: Study corridor (Courtesy: Google Maps ®) 11
Figure 5: Flowchart for Vissim COM driver script. 13
Figure 6: Architecture for dynamic visualization of energy performance indicators. 14
Figure 7: Energy plot generated dynamically during simulation runtime. 15
Figure 8: Routes selected for studying model's sensitivity to real-time input. 16
Figure 9: Average vehicle travel time versus simulation time intervals plots for (a) Westbound Route 4, and (b) Eastbound Route 6. 17
Figure 10: (a) Scatterplot for energy consumption versus travel time for Route 3, and (b) Cumulative energy consumption density function for vehicles on Route 3. 18
Figure 11: Study corridor-2.3 miles of North Avenue Smart Corridor, including 15 signalized intersections. 21
Figure 12: Updated real-time data-driven simulation model architecture overview. 21
Figure 13: Snapshot of raw volume count data. 22
Figure 14: Snapshot of raw signal event data. 23
Figure 15: Overview of *dynamic data-driven traffic simulation* initialization logic. 24
Figure 16: Overview of *dynamic data-driven traffic simulation* runtime logic. 25
Figure 17: Sample from signal state data table 27
Figure 18: Sample from volume data table 27
Figure 19: Demarcation process for segments 32
Figure 20: Association of data points to a segment 33
Figure 21: Sample KPI visualization lookup-table 35
Figure 22: Screenshot of visualization website for March 18, 2019 36
Figure 23: Four-steps of data processing to obtain the *Standardized Name Table* from *Raw Volume Table* 38
Figure 24: 3D visualizations of missing data intervals by hour, with 24-hours of a day on the x-axis, 112 days on the y-axis, and detectors on the z-axis. Figures (a-k) show hours with 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, or 10 6-minute missing intervals. 39
Figure 25: Interactive 2D visualizations showing intermittent missing data patterns, aggregated over days: (a) 1-28, (b) 29-56, (c) 56-84, and (d) 85-112. 40
Figure 26: Missing volume pattern for 147 detectors over 24 hours on February 15, 2019. 41
Figure 27: Volume detection availability at the intersections, per lane, per movement. Permanently missing movements are identified in bold and red. 42
Figure 28: All-RED irregularities: (a) Peachtree St. NE and (b) Parkway Dr. NE 43
Figure 29: RED duration irregularities 44
Figure 30: AMBER duration irregularities: (a) Piedmont Ave., (b) Peachtree St. NE 45
Figure 31: (a) *Base day* raw volume data stream to *base day* with imputed data, (b) Volume imputations for *raw data Streams processing module.* 46
Figure 32: Elbow method to suggest K value. Sum of squared distances versus number of clusters. 50

Figure 33: Two clusters of 24-hour data gap patterns with dark blue blocks as data loss intervals. 51
Figure 34: Clusters of data points projected in the 3-PC space........ 51
Figure 35: Missing volume data pattern generation methodology......... 52
Figure 36: Missing volume Pattern 1 generated for the sensitivity experiment......... 54
Figure 37: Routes on which vehicle travel times are studied for Pattern 1 detector outages....... 58
Figure 38: Boxplot of 85th percentile vehicle travel times for data imputation error cases, for Pattern 1 mainline and side street routes........ 59
Figure 39: Mean simulation vehicle input count for data imputation error cases in comparison to the *base case* on the side-street routes that observed permanent data loss in Pattern 1. 59
Figure 40: Variation in 85th percentile travel time values at the mainline and side street routes for the five missing volume patterns. 60
Figure 41: Variation in average 85 percentile travel time differences *with intermittent loss scenario* and *without intermittent loss scenario* of all time bins for different error in data imputation cases. 63

Table of Tables

Table 1: Intersection IDs........ 28
Table 2: Eigenvector matrix vector representation of the first three principal components........ 48
Table 3: Summary of Boundary and Internal Approach Lanes with Permanent or Intermittent Data Loss for the Five Detector Outage Patterns. 55
Table 4: Average 85th percentile difference of all time bins for 10 random seed runs for the three error imputation scenarios........ 62

1. INTRODUCTION

According to the 2015 Urban Mobility Scorecard (TTI, 2015), traffic congestion caused drivers to waste more than 3.1 billion gallons of fuel and cost travelers nearly 7 billion extra hours behind the wheel (42 hours/commuter/year); the total nationwide price tag is about $160 billion, or $960 per commuter. Congestion impacts also go well beyond immediate travel; for instance, it is predicted that public health impacts will increase to $17 billion by 2030 (Levy, Buonocore et al. 2010). Improving the efficiency and utilization of existing facilities, especially major and minor arterials, through intelligent utilization of current infrastructure has become paramount as resources have become increasingly constrained and the cost of new capacity has become prohibitive.

Through its Smart City initiatives the City of Atlanta (CoA) is tackling these challenges head-on, rapidly becoming a national leader in Smart Cities, integrating data, communications, analytics, and implementation in the intelligent and informed management of our transportation system. This project presents activities designed to demonstrate intelligent transportation system (ITS) technologies and congestion mitigation in Atlanta on the North Avenue Smart Corridor, where advanced vehicle-to-vehicle (V2V) and vehicle-to-roadside (V2R) technologies have been deployed (RenewAtlantaBond 2018).

1.1. North Avenue ITS Demonstration Testbed Overview

A key component of the ongoing CoA Smart City initiative is the implementation of traffic operational improvements designed to minimize key surface transportation performance measures, such as travel time and congestion. In collaboration with the already extensive efforts underway by the CoA, the Georgia Tech team seeks to further enhance the ongoing efforts though a hybrid implementation of data collection, data integration, data analytics, and advanced simulation modeling. Georgia Tech is utilizing the North Avenue corridor as a "Green Corridor" demonstration site. We are developing a dynamic data-driven system which employs real-time operations data (Dedicated Short-Range Communications (DSRC), Bluetooth, signal control, etc.) from the North Avenue corridor integrated monitoring system. The basic conceptual approach includes:

- Utilizing CoA sensor infrastructure to monitor near-real-time high-resolution corridor level conditions from which traffic, energy, and emissions behaviors will be assessed via big data applications.
- Developing an advanced simulation model (Vissim) of the North Avenue Corridor capable of representing detector, signal, DSRC, and other sensor data.
- Applying big data processes to integrate streaming signal timing and vehicle movement data to assess emissions and energy usage for simulated and field data.
- Visualizing Green Corridor energy and emissions data.
- Delivering useful Key Performance Indices (KPIs) to system operators to inform updates in network control (e.g., optimize corridor-level energy consumption and emissions).

- Utilizing the developed simulation to determine the potential for energy and emission related KPI improvements as the traveling public begins to widely adopt connected vehicle (i.e. DSRC) and/or other Smart City technologies.

While not directly addressed in this initial effort, this project lays the foundation for further advancements in real-time, dynamic, data-driven Smart City applications. Through such an effort it will be possible to: integrate online dashboards for system monitoring and operations; close the control loop with tailored messages direct to individual travelers to reduce emission and energy usage; support real-time optimization of signal control strategies balancing travel time, throughput, emissions, and energy usage; integrate predictive analytics for near-future optimization; and expand the system beyond North Avenue.

1.2. Project Background

Smart City programs are fast becoming a reality (Rouse 2017, U.S. DOT n.d.)(Rouse 2017; U.S. DOT n.d.). A Smart City uses information and communication technology (ICT) facilities to develop insights to improve the quality of life of residents, workers, and visitors (U.S. DOT 2014) (U.S. DOT 2014). Within the transportation context Smart City initiatives emphasize technology deployment, with roadways and vehicles being equipped with connected vehicle (CV) technologies and numerous new embedded sensors. Such a Smart City is expected to consume real-time information and leverage communication technologies to address urban challenges, including transportation related issues such as traffic congestion and environmental pollution. Addressing these challenges entails the consumption of high volumes of high velocity data to influence real-time decisions in an effort to optimize the use of transportation systems and services. The early major incentive for the deployment of many CV and embedded sensor technologies was the potential for reductions in vehicle crash related injuries and fatalities (U.S. DOT 2017, U.S. DOT n.d.) (U.S. DOT 2017; U.S. DOT n.d.). However, with high-speed data transactions in a connected environment, travelers and traffic management centers are expected to receive relevant information in real-time, which will not only enable safer travel, but also enhance environmental friendly route/mode choices and optimized traffic operations.

Smart connected vehicle corridors are envisioned to use wireless communications such as DSRC or 5G, on-board computer processing, advanced vehicle sensors, GPS navigation, and smart infrastructure to build a connected network. However, as CV is still an emerging technology, with no requirement from either the Department of Transportation or automobile manufacturers to deploy infrastructure or on-board units, research is limited by the availability of testbeds. In addition, there is currently a frequent arrival of new embedded sensors being developed for smart city corridors. It is expected that as these technologies are tested, and with the incentives provided by smart cities initiatives, there will be incremental adoption of the technologies in the field. During this transition phase, as the technologies are deployed there will be a mixed or hybrid environment in which the equipment, vehicles, and drivers will have to operate until there is a full saturation of technology adoption. For investigating the effective application of additional data available through CV and embedded sensor technologies and to prepare for future deployments or modification of operations, it is critical to develop the capability to simulate this hybrid infrastructure environment so that different scenarios can be developed and tested.

To accomplish such development and testing the City of Atlanta has selected the North Avenue Corridor as an example Smart City transportation implementation, championed by the Renew Atlanta program. The project aims to make the North Avenue NW Corridor a smart connected corridor by integrating data, communications, and analytics in the intelligent and informed management of the transportation system (RenewAtlantaBond 2018). The North Avenue corridor in Atlanta has been subject to rapid changes in land use and transportation infrastructure. The development of Ponce City Market and the Beltline bicycle and pedestrian trail are already affecting travel patterns and corridor mobility. Resulting congestion increases energy consumption and vehicle emissions for the 29,000 households living in the 14 surrounding neighborhoods as well as other CoA residents that utilize the corridor. The potential development of a Beltline streetcar line and automated transit demonstration project on North Avenue provides significant additional incentive to establish a technology testbed in this corridor and monitor the evolution of travel patterns and congestion over time. Figure 1 and Figure 2 illustrate the corridor and surrounding neighborhoods.

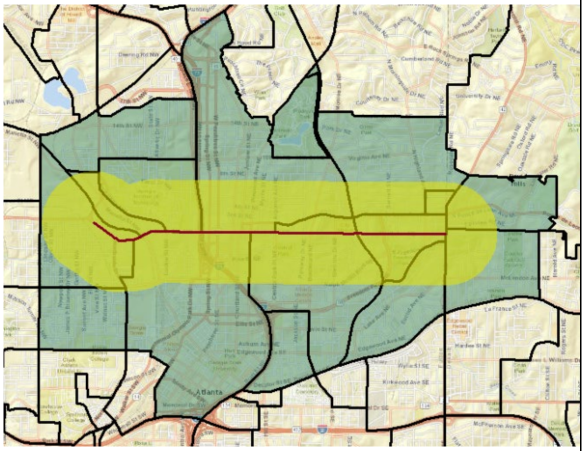
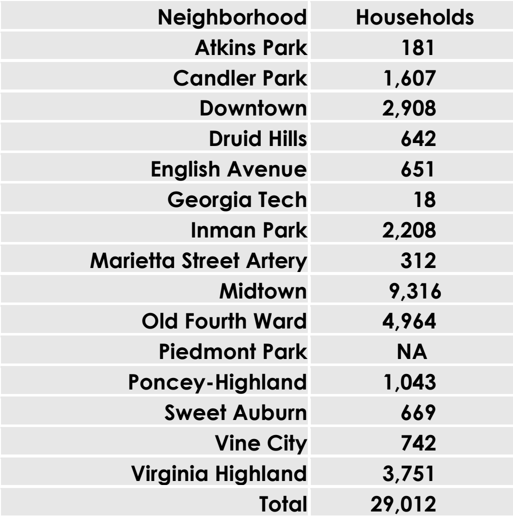


Figure 1: North Avenue corridor and local neighborhoods



Neighborhood	Households
Atkins Park	181
Candler Park	1,607
Downtown	2,908
Druid Hills	642
English Avenue	651
Georgia Tech	18
Inman Park	2,208
Marietta Street Artery	312
Midtown	9,316
Old Fourth Ward	4,964
Piedmont Park	NA
Poncey-Highland	1,043
Sweet Auburn	669
Vine City	742
Virginia Highland	3,751
Total	29,012

Figure 2: Households in each North Avenue corridor neighborhood

The research team model development occurred in two major phases, the first was the development of a hybrid model with fifteen field calibrated signalized intersections on 2.6 miles of the North Avenue Smart corridor. Within this hybrid model two of the signals are driven by real-time signal status and vehicle arrival data streams, the remaining intersection's volumes and signal control are driven by the simulation. Chapter 3 of this report discusses the architecture development and implementation required to create this hybrid model. This includes the integration of multiple physically separated systems, to enable communications, drive the simulation, and generate performance metrics in real-time. To evaluate the robustness of the approach, simulation results and performance metrics from the real-time data injection scenario are compared to the control case where all inputs are predetermined.

The second major phase of the project, presented in Chapter 4, updates the model architecture to incorporate streaming data from all 15 intersections. During this phase of the research it was seen that continuity in retrieval of high frequency data from a connected corridor can be interrupted due to communication or other challenges. Such data gaps in volume count or signal data streams can lead to erroneous simulation results. The error can be more pronounced if the gap duration is long and occurs at a crucial corridor location, where the volume or signal has a significant influence on corridor operations. Presence of such data gaps in the data streams that drive the simulation model can lead to a significantly different emulation of the traffic state in the simulation versus that in the field. Thus, the second phase of the project also uses the developed connected corridor data-driven simulation model to derive insights on the impact of data loss and potential error is data imputation on performance measures produced from the simulation model. This effort allows for recommendations related to the development of a production level model.

Finally, under the Georgia Institute of Technology Smart Cities Project a sidewalk, ramp, and curb cut inventory and condition assessment was conducted for Renew Atlanta for several corridors in Atlanta. As these inventories and assessments have been separately reported to the Renew Atlanta project team they are not discussed within this report. However, a brief summary of the effort is provide in Appendix A.

2. LITERATURE REVIEW

This section reviews past efforts in the development and application of real-time data driven traffic simulation models. The review focuses on three main aspects: 1) connected corridor deployments and studies, 2) real-time traffic data driven simulations, and 3) traffic data imputation methodologies.

2.1. Connected Corridor Deployments and Testbed Studies

Cities across the world are undertaking smart city initiatives to improve resident and visitor quality of life, increase sustainability, and efficiently utilize resources. While there is no one smart city definition (Greco 2015), smart city concepts generally focus on advancing urban services by leveraging technology (Glasmeier A 2015, Williams 2018). A common smart city transportation initiative is the deployment of sensors and advanced technologies to create *smart connected corridors*. For example, the Pennsylvania Department of Transportation (PennDOT) aims to use a coordinated smart corridor system to reduce congestion and crashes, and to provide real-time traffic information (PennDOT 2018). The Tennessee Department of Transportation (TDOT) is focusing on upgrading signals to optimize intersection operations on the Interstate 24 smart corridor (TDOT n.d.). In England, the A2M2 connected corridor between London and Dover is being used as pilot testbed to explore information transference between vehicles and infrastructure. The corridor is expected to provide enhanced mobility services, such as real-time traffic management, as well as incident response through real-time personalized connectivity with the drivers (Seymour 2018). The Minneapolis Department on Transportation (MnDOT) has selected part of the Highway 55 corridor for connected vehicle technology deployment. Using this corridor as a testbed, MnDOT aims to test connected infrastructure and develop a data management system to support information sharing across connected environment stakeholders. The MnDOT connected corridor project aims to aid traffic safety by providing real-time assistive information to drivers about work zones, lane merges, pedestrian conflicts, etc. (MnDOT n.d.). Similar connected corridor projects are in progress in several other U.S. states, including New York, California, Virginia, Wyoming, and Georgia (Caltrans , CityOfAtlanta 2017, WYDOT 2017, NYCDOT 2019, UCBerkeley n.d., VTTI n.d.).

To achieve the vision of leveraging information sharing between vehicles and the infrastructure in order to enhance safety, mobility, and sustainability of transportation services, it is imperative to test different aspects of connected vehicle (CV) technologies under realistic field conditions. To better understand and prepare for the "technical, institutional, and financial challenges" that emerging CV technology deployment will bring, the United States Department of Transportation (USDOT) is supporting the on-road deployment and testing of these technologies (USDOT n.d.). As part of this program, sites in New York, Wyoming, and Tampa, are testing CV applications that cater to local transportation needs. The connected vehicle systems at these sites have been operational since fall 2018. The impact on key performance measures at these locations are to be studied at least until fall 2021. The New York City Department of Transportation (NYCDOT) pilot is using a selected number of vehicles fitted with CV technology and roadside units (RSUs) deployed at selected intersections to test vehicle-to-infrastructure

(V2I) safety applications. In addition, at this site, in-vehicle applications to avoid vehicle-pedestrian conflicts will also be tested. The Tampa site focuses on testing V2I and vehicle-to-vehicle (V2V) safety and mobility applications, such as an intelligent traffic signal system. The key focus of the Wyoming Department of Transportation (WYDOT) is to use V2I and V2V applications to address challenges associated with weather and freight movement. For this, WYDOT is developing advisories to provide safe transportation and an efficient incident management systems, such as roadside alerts, parking notifications, and dynamic travel guidance (USDOT n.d.). Availability of high frequency data in such a smart corridor can be used to provide more accurate real-time dynamic information to road users and to transportation organizations.

Research on developing the necessary tools to test connected applications to quantify the impacts of this new technology is ongoing. For example, to evaluate the performance of a connected application - cooperative adaptive cruise control (CACC), Zulkerfli et al. (2016) recognized the need for a tool to test connected applications in a safe environment. The study built a simulation model to conduct an experiment with hypothetical input values for the number of vehicles in a network. The experiment was designed considering a mixed fleet of connected and non-connected vehicles (Mohd Zulkefli, Mukherjee et al. 2016). Virginia DOT derived an algorithm for real-time prediction of vehicle locations in a connected environment using a traffic simulation model. The algorithm estimated locations of non-connected vehicles in the network using locations of the connected vehicles and other traffic parameters. The study also tested a connected application to optimize signal timing (Goodall 2013). Doecke et al., studied a V2V safety application of the market available connected vehicle technologies. In this study vehicle trajectory data from simulated crash scenarios are input to an On-Board Unit (OBU) that generated Basic Safety Messages (Doecke, Grant et al. 2015). Use of such hardware-in-the-loop (HITL) simulation platforms, are expected to advance the modeling of complexity found in connected environment systems (U.S. DOT 2016, U.S. DOT 2017). Chowdhary et al., studied connected applications to provide insights on handling real-time connected data streams and big data management tools (Chowdhury, Rahman et al. 2018). The USDOT program Applications for the Environment: Real-Time Information Synthesis (AERIS) studied connected environmental applications. Applications such as eco-approach and departure at signalized intersections, eco-traffic signal timing, and eco-traffic signal priority, etc., are tested on a simulation model of a 27 intersection network. The study utilizes historic volumes and signal data and different penetration rate of connected vehicles in the fleet. In this study, connected technology data such as Signal Phasing and Timing (SPaT) data distributed by the road side units (RSUs) and on-road vehicle information such as location, speed, etc., provided by Basic Safety Messages (BSM) are simulated assuming a 100% penetration rate of RSUs (U.S. DOT 2014).

2.2. Real-Time Traffic Data-Driven Simulation Modelling

The integration of smart technologies such as sensors, networked communications, and hardware and software computing with the physical infrastructure is central in creating a Smart City (Brügmann, Schreckenberg et al. 2013, U.S. DOT 2014, Pop and Proștean 2018). One area of significant focus is the integration of the new technologies enabling an improved estimation of the traffic state and real-time traveler and traffic information

(Allström, Barcelo et al. 2017, Chen and Du 2017). Given the inherent challenges in field experimentation, the use of traffic simulation models driven by real-time input data to emulate the real-world environment is utilized in numerous studies (Xiaowen, Ferman et al. 2003, Brügmann, Schreckenberg et al. 2013). Various traffic simulation tools have been used by researchers, depending on the modeling requirements. A University of Leeds report in 2000 compared the capability of several macroscopic and microscopic simulation models in developing real-time traffic management solutions ("SMARTEST" 2000). More recently, a similar study looked into 17 simulation software tools and noted "a lack of online traffic simulation software applications specially designed for heterogeneous road transportation networks" (Pell, Meingast et al. 2017).

The concept of using real-time data to drive a traffic simulation model is not new. Previous studies, such as Henclewood et al. (2010), injected real-time vehicle detection data into microscopic simulation models to simulate the current traffic state, which was then used to generate predictions of future traffic states (Henclewood, Guin et al. 2010). An early example of real-time simulation is provided by Maroto et al., who developed a microscopic model that could simulate driving simulator traffic scenarios in real-time, where vehicle behavior model is based on car following theory (Maroto, Delso et al. 2006). An effort, tested in the Dutch city of Assen, built a real-time traffic model that used traffic flow and travel-time data from different sensor technologies such as cameras, highway loop detectors, and Bluetooth® sensors, to predict the short-term traffic state. The model architecture connected the real-time traffic measurements with the macroscopic dynamic traffic assignment model "StreamLine". Traffic counts were used for model calibration and forecasting (Wismans, de Romph et al. 2014). A study by Sturari, Catani et al. presented the use of in-field mobile and fixed sensor data to drive a real-time microscopic traffic simulation model built using the Simulation of Urban Mobility (SUMO) simulation package. Model input included the real-time traffic count and vehicle location data obtained from different sources such as induction loop, camera counter, radar counter, automatic vehicles location (AVL) systems, etc. (Sturari, Catani et al. 2016). In these studies, the real-time data comprised of vehicle detection or vehicle position data; however, infrastructure information such as the traffic signal state, state of ramp meters, information from variable speed limit signs, etc., were assumed to be pre-encoded in the simulation model based on known logic or field calibration. Today, with the richness of information in a smart city CV environment, where the vehicle sensor data may result in real-time changes to the signal control, ramp meter rates, etc., it is imperative that the state of the infrastructure is updated in the simulation to ensure the accuracy of the simulation results. Anthony et al. studied computational benefits of using parallel processing for simulating traffic flows in a real-time traffic simulation system and concluded that a parallel system can be crucial for real-time traffic simulations (A. Chronopoulos and Johnston 1998). USDOT Federal Highway Administration (FHWA) is using the hardware-in-the-loop approach, in which the connected infrastructure and connected vehicle data are fed as input to the simulation model, to study the application of CV technologies. The research is expected to provide engineers with a technology that will allow CV applications engineers to test various simulation scenarios and obtain meaningful results (FHWA 2019).

The availability of these various data streams also provide interesting challenges. The volume, velocity, and wide variety of these data streams naturally suggest the use of big data technologies for extracting useful information. Previous studies such as Amini et al. (2017) have used tools such as Kafka to address the issues of volume and variety of big data (Amini, Gerostathopoulos et al. 2017). Lv et al. (2015) used deep learning techniques for predicting traffic flow (Lv, Duan et al. 2015). The current study takes a hybrid approach, where the architecture allows for the use of big data concepts in the extract-transform-load (ETL) stages preceding the injection of the data into a simulation model.

2.3. Traffic Data Imputation Methodologies

The presence of gaps in traffic data is common and, thus, traffic data imputation methodologies have long been of interest to traffic engineers. The collected traffic data can be volume count data, vehicle speed data, or one of many other types of data. Heuristic imputing methods are used, such as replacing missing values by another day's data directly, or by using an average, moving average, or weighted moving average of historical data (Zhong, Sharma et al. 2005, Roh, Canadá et al. 2016). In 2006, Zhong et. al applied a pattern matching-based method which chooses the day for replacement values based on the match of values obtained in previous selected stretch of hours (Zhong and Sharma 2006). Auto-regressive integrated moving average (ARIMA) models, in which missing data are predicted based on the available preceding values, have been used and studied in several efforts (Lee and Fambro 1999, Williams 2001). ARIMA-based models have been frequently used with modifications to account for long-term seasonal variations, i.e., Seasonal ARIMA (Ghosh, Basu et al. 2005, Spławińska 2015, Pell, Meingast et al. 2017), and to account for spatial variations, that is, space-time ARIMA (Ding, Wang et al. 2010). In the learning methods, the non-parametric learning k-nearest neighbor (kNN) algorithm for imputations has shown high accuracy (Oehmcke, Zielinski et al. 2016, Tak, Woo et al. 2016, Sun, Ma et al. 2017). A recent study by Zhuang et al. applied an image inpainting approach based on a convolutional neural-networks (CNN) to find values for imputation of missing volume data. In comparison to two other methods (Bayesian principal component analysis and denoising stacked auto encoder, which is a deep learning based approach), the CNN-based approach performed better (Zhuang, Ke et al. 2019). As in the initial step, this study considers a simple heuristic volume data imputation approach.

3. PHASE 1: HYBRID MODEL ARCHITECTURE, DATA DESCRIPTION, AND PERFORMANCE TEST

This chapter presents an assessment of the feasibility aspects of using a real-time data-driven transportation simulation model to evaluate and visualize network performance indices to provide dynamic operational feedback in a real world environment, in a big data context. A hybrid traffic simulation model, consisting of a mix of preset and real-time data-driven intersections, is developed. The hybrid model represents a traffic corridor partially equipped with smart devices generating high velocity, high volume datasets with limited a shelf-life. The model used in this study emulates seventeen consecutive intersections on a corridor. Signal controls and vehicle volumes at two of the intersections are driven by real-time data while the remaining intersections are driven by preset data. An optimized architecture is developed to enable control of the signals and the vehicle volumes using real-time data from in-field detectors, and real-time processing of the vehicle trajectories from the simulation output to generate travel-time, energy, and emissions performance indices.

3.1. Introduction

A smart city is expected to consume real-time information and leverage communication technologies to address urban challenges, including transportation related issues such as traffic congestion and environmental pollution. An example smart city transportation implementation is currently being championed by the Renew Atlanta program, Atlanta, Georgia, USA, as part of the North Avenue Smart Corridor project. The project aims to make the North Avenue NW Corridor a smart connected corridor by integrating data, communications, and analytics in the intelligent and informed management of the transportation system (CityofAtlanta 2018). This chapter presents a hybrid real-time data-driven transportation microscopic simulation model of 2.6 miles of the North Avenue Smart corridor that allows near-real-time visualization of network performance metrics.

As discussed, smart connected vehicle corridors are envisioned to use wireless communications such as DSRC or 5G, on-board computer processing, advanced vehicle sensors, GPS navigation, and smart infrastructure to build a connected network. However, as CV is still an emerging technology, with no requirement from either the Department of Transportation or automobile manufacturers to deploy infrastructure or on-board units, research is limited by the availability of testbeds. In addition, there is currently a frequent arrival of new embedded sensors being developed for smart city corridors. It is expected that as these technologies are tested, and with the incentives provided by smart cities initiatives, there will be incremental adoption of the technologies in the field. During this transition phase, as the technologies are deployed, it is expected that there will be a mixed or hybrid environment in which the equipment, vehicles, and drivers will have to operate until there is a full saturation of technology adoption. For investigating the effective application of additional data available through CV and embedded sensor technologies and to prepare for future deployments or modification of operations, it is critical to develop the capability to simulate this hybrid infrastructure environment so that different scenarios can be developed and tested. Such a hybrid model that simulates an incremental deployment scenario where connectivity to the edges is partial, is presented here.

The model contains seventeen field calibrated signalized intersections with two of the signals driven by real-time signal status and vehicle arrival data streams. This study develops and implements the architecture required to integrate multiple physically separated systems, to enable communications, drive the simulation, and generate performance metrics in real-time. To evaluate the robustness of the approach, simulation results and performance metrics from the real-time data injection scenario are compared to the control case where all inputs are predetermined.

3.2. Model Architecture

To capture the intricacies of the incremental deployment of CV technologies and infrastructure connectivity, this study proposes the use of a hybrid simulation that is partially modeled using preset data and partially modeled using real-time data. This requires the integration of multiple disparate modules. There are three high-level tasks 1) real-time data injection into the simulation, 2) simulation model execution, and 3) generation of performance metrics. Dynamic links are established between these three components to ensure faster than real-time computations, ensuring that the simulation does not fall behind the real-world operations. Figure 3 presents the complete model architecture.

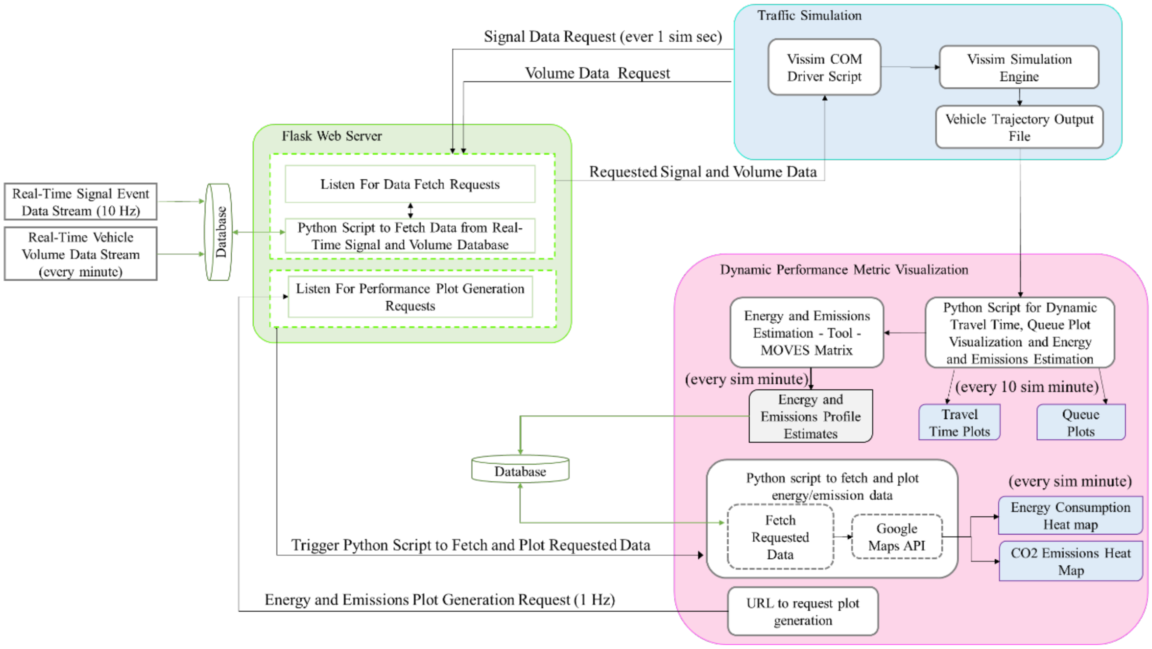

Traffic Simulation
Signal Data Request (ever 1 sim sec)
Vissim COM Driver Script
Vissim Simulation Engine
Volume Data Request
Vehicle Trajectory Output File
Flask Web Server
Listen For Data Fetch Requests
Requested Signal and Volume Data
Real-Time Signal Event Data Stream (10 Hz)
Real-Time Vehicle Volume Data Stream (every minute)
Database
Python Script to Fetch Data from Real-Time Signal and Volume Database
Listen For Performance Plot Generation Requests
Dynamic Performance Metric Visualization
Energy and Emissions Estimation - Tool - MOVES Matrix
Python Script for Dynamic Travel Time, Queue Plot Visualization and Energy and Emissions Estimation
(every sim minute)
(every 10 sim minute)
Energy and Emissions Profile Estimates
Travel Time Plots
Queue Plots
Database
Python script to fetch and plot energy/emission data
(every sim minute)
Fetch Requested Data
Google Maps API
Energy Consumption Heat map
CO2 Emissions Heat Map
Trigger Python Script to Fetch and Plot Requested Data
Energy and Emissions Plot Generation Request (1 Hz)
URL to request plot generation

Figure 3: Hybrid model architecture.

Central to the architecture is the Flask micro web framework that acts as the communication facilitator and data broker. Several PERL and Python scripts perform the task of continuously fetching the data from the sensors on the field and populating a database. The microscopic simulation requests the data from this database via requests through the Flask engine and receives data as and when it becomes available. The simulation writes the vehicle trajectories to an output file which is scanned continuously by another set of Python scripts that generate the performance metrics. The results are available for retrieval via requests through the Flask interface. Details of the major components and the links between them are discussed in the following subsections.

3.2.1. Traffic Simulation Model

The traffic simulation model is built using an off-the-shelf simulation package, PTV's Vissim version 9.00-08. Primary inputs for the Vissim transportation model are signal timing plans, vehicle volume inputs and turning movement percentage splits at the intersections. The signal timing plans of the intersections on the corridor were obtained from Renew Atlanta staff. The vehicle volume inputs and turning ratios were obtained by manual vehicle count data collection at the intersections. At two consecutive intersections, Glen Iris Drive at North Avenue and Ponce City Market at North Avenue, signal controls and vehicle volumes are driven using real-time emulation of signal control events and vehicle volume data from the field sensors. The remaining 15 intersections use the preset values in the calibrated model. Figure 4 shows a snapshot of the network corridor with the two real-time data-driven intersections highlighted.

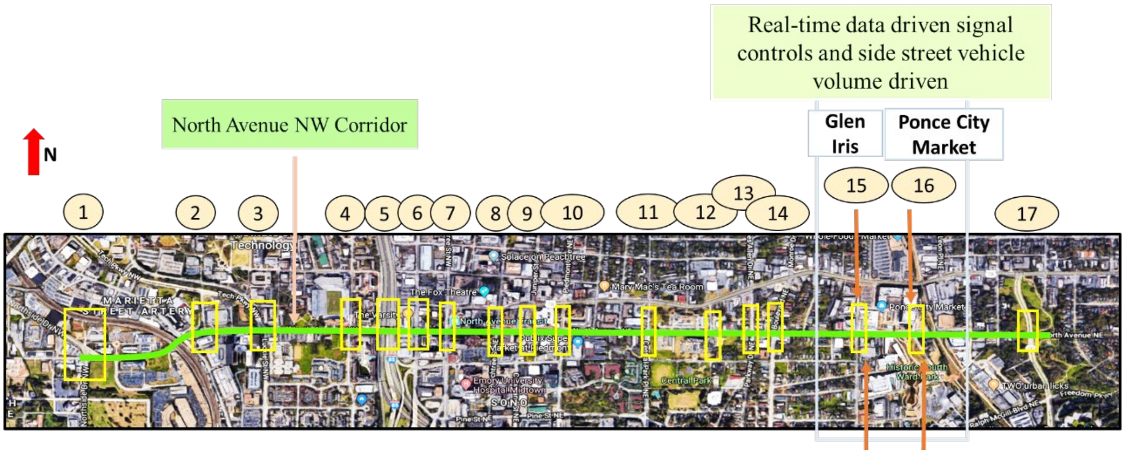


Figure 4: Study corridor (Courtesy: Google Maps ®)

3.2.1.1. Volume Update Logic

The vehicle volume simulation inputs for the northbound and southbound approaches at the two data driven intersections (Glen Iris Dr. and Ponce City Market) are updated every 5 minutes to match the fluctuations in real-time vehicle volume. This is implemented based on the previously field collected initial flow data used in the model calibration, adjusted to reflect the 5 minute interval real-time proportional fluctuations in the traffic volume. This approach allows the vehicle volume inputs at the two real-time data driven intersections to remain consistent with the calibration field-collected corridor volumes.

3.2.2. Injecting Real-Time Intersection Signal State and Volume Count Data

The volume and signal state input data are obtained from sensors in the field. The data are polled out of the field sensors by a set of PERL scripts and stored in a database. The data are injected into the simulation during runtime using Vissim's COM interface to emulate the near-real-time state of signal control and vehicle volume for the two intersections in the simulation model. The Flask server provides the interface for Vissim COM to fetch the data from the database.

3.2.2.1. Signal Controller Data Parsing

The signal events data that is polled out of the field signal controllers contains a 10Hz series of timestamped eventIDs and the corresponding parameters. The events of interest required to drive the signal controllers in the simulation model are:

- 1: Start of Green Indication
- 8: Start of Amber Indication
- 10: Start of Red Indication

For these Event Type IDs, the parameter column provides corresponding signal phase number. Additional detail on streaming sensor data is provided in Chapter 4.

3.2.2.2. Vissim COM-Interface

The Vissim COM interface is used to change the simulation network objects during runtime based on the signal phase and sensor inputs. Python 2.7 is used to drive the Vissim simulation model using its COM interface, by running the simulation model in single step mode, and changing the signal controls and vehicle volumes during run time at the respective time steps. The simulation resolution is set to 10; that is, 10 simulation time steps are executed per simulation second. For every time step of the simulation period, a series of COM commands are executed along with the command to run the next simulation step.

At every 10th time step that marks the start of a simulation second, the COM driver script requests intersection signal event data for the corresponding timestamp. The driver script then assesses the fetched data to determine whether the event type ID for a signal phase indicates a change of state such as start of green, start of amber, or start of red indication, and updates the state of the corresponding signal head object in the simulation.

To update the vehicle volume-input for the two data driven intersections, the COM script sends a request every 3000th time step (5th simulation minute) to fetch the vehicle volume for the last 5 minutes. The input volumes are determined using the logic described earlier. The COM script's flowchart to execute each simulation time step is shown in Figure 5, with the requests for signal and volume data marked as callouts 1 and 2.

3.2.3. Dynamic Performance Evaluation Visualizations

The simulation's output is captured in a vehicle trajectory output file which updates during runtime. The performance evaluator module fetches data from the Vissim trajectory output file, with non-locking reads, and performs the computations for energy and emissions. The visualization of these parameters in the form of time series plots, available in real-time.

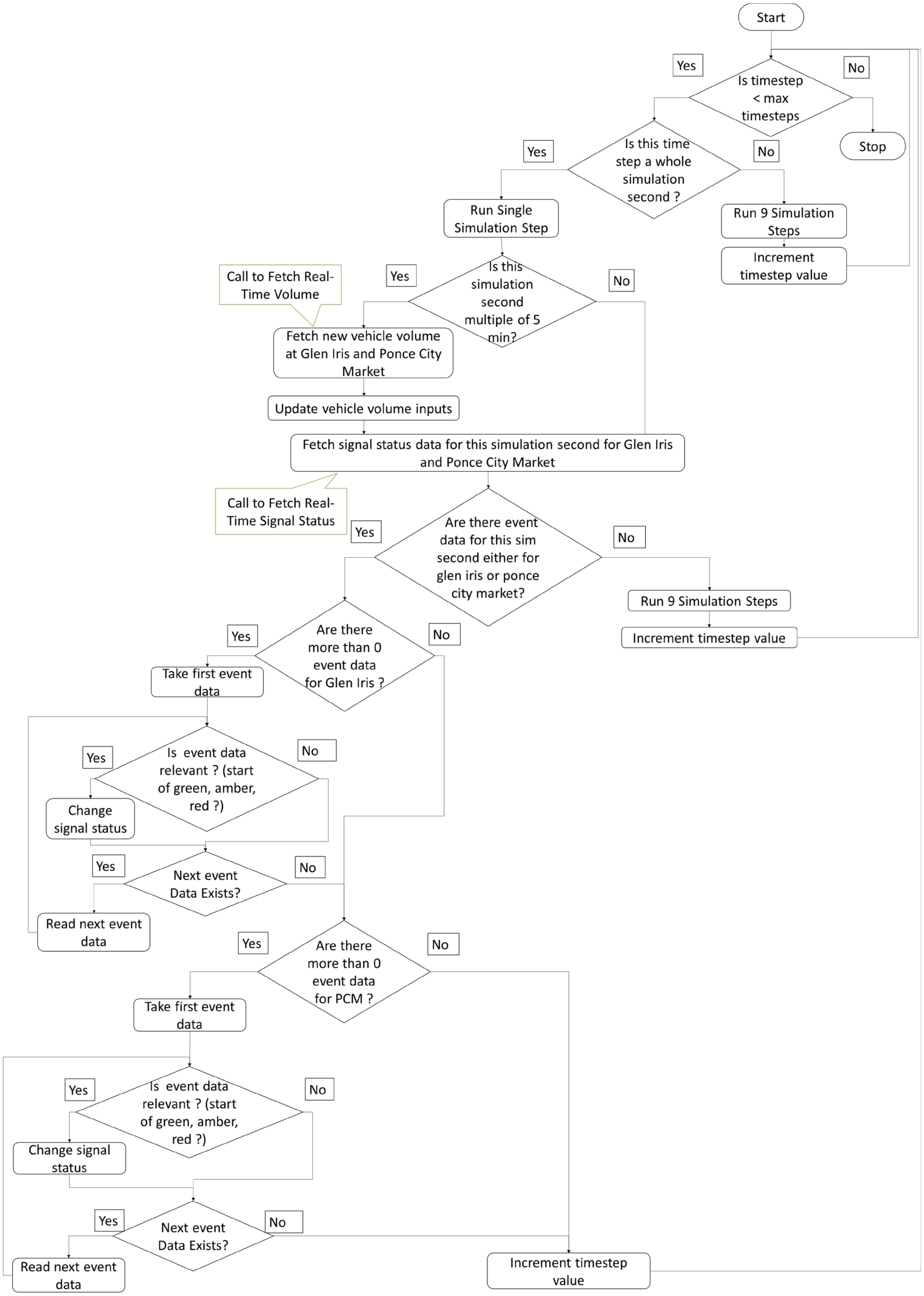

Start
Is timestep < max timesteps
Yes
No
Stop
Is this time step a whole simulation second ?
Yes
No
Run Single Simulation Step
Run 9 Simulation Steps
Increment timestep value
Is this simulation second multiple of 5 min?
Yes
No
Call to Fetch Real-Time Volume
Fetch new vehicle volume at Glen Iris and Ponce City Market
Update vehicle volume inputs
Fetch signal status data for this simulation second for Glen Iris and Ponce City Market
Call to Fetch Real-Time Signal Status
Are there event data for this sim second either for glen iris or ponce city market?
Yes
No
Run 9 Simulation Steps
Increment timestep value
Are there more than 0 event data for Glen Iris ?
Yes
No
Take first event data
Is event data relevant ? (start of green, amber, red ?)
Yes
No
Change signal status
Next event Data Exists?
Yes
No
Read next event data
Are there more than 0 event data for PCM ?
Yes
No
Take first event data
Is event data relevant ? (start of green, amber, red ?)
Yes
No
Change signal status
Next event Data Exists?
Yes
No
Read next event data
Increment timestep value

Figure 5: Flowchart for Vissim COM driver script.

3.2.3.1. Energy-Emission

The trajectory output file is also used to estimate energy and CO_2 emissions profile in real-time at a predetermined regular interval using the Motor Vehicle Emission Simulator (MOVES) matrix. MOVES matrix is a computationally improved version of the energy estimation tool MOVES, developed and mandated by the US Environmental Protection Agency (USEPA) (Guensler, Liu et al. 2017). It estimates energy and emissions for off-road and on-road vehicles based on the vehicle type, weather conditions, and vehicle model, make, and year. For energy and emissions estimates, the 10 Hz data in the trajectory output file is too noisy for direct use. Thus, the 10 Hz data is condensed to 1 Hz by taking the median of every 10 records for every vehicle.

The vehicles' coordinates are used to assign the vehicles to 200-ft segments on the network (discussed further in Chapter 4). A visualization of the cumulative energy consumption per segment for fixed intervals, such as 60 seconds, is created using the Google Maps API. Energy and CO_2 emission heat maps refresh automatically at fixed intervals (say every 60 seconds) synchronous with the simulation run. Figure 6 depicts the described architecture used for dynamic visualization of the energy performance metrics.

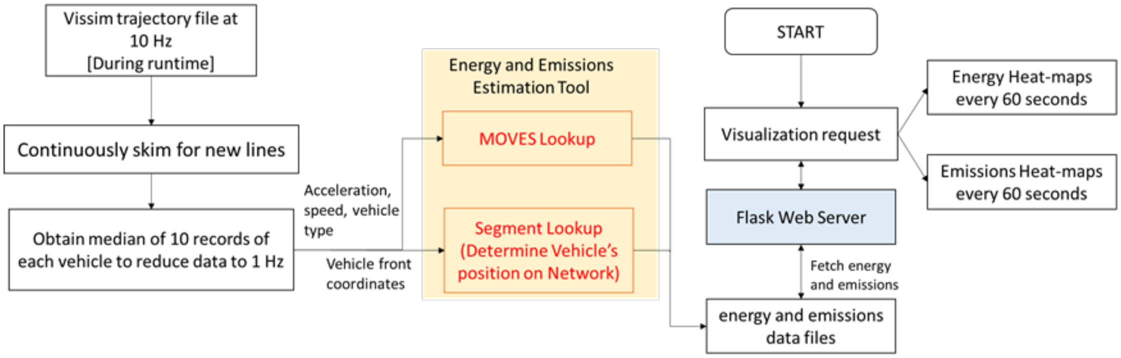

Vissim trajectory file at 10 Hz [During runtime]
Continuously skim for new lines
Obtain median of 10 records of each vehicle to reduce data to 1 Hz
Acceleration, speed, vehicle type
Vehicle front coordinates
Energy and Emissions Estimation Tool
MOVES Lookup
Segment Lookup (Determine Vehicle's position on Network)
START
Visualization request
Flask Web Server
Fetch energy and emissions
energy and emissions data files
Energy Heat-maps every 60 seconds
Emissions Heat-maps every 60 seconds

Figure 6: Architecture for dynamic visualization of energy performance indicators.

3.3. Result and Discussion

The results presented here are two fold. The first set of results refer to the real-time outputs from the model during runtime. The second set presents the results of the test of robustness of the simulation model in the presence of real-time inputs and the test of sensitivity of the model to these real-time inputs. Due to the need for multiple iterations of the model run (with exact same input conditions) for the comparative analysis, the experiments were conducted with pre-fetched input data.

3.3.1. Real-time Performance Measures Computation

Near-real-time traffic information visualized for performance measures such as queue lengths, energy consumption, and CO_2 emissions can be used in traffic operations to interpret and update eco-driving routes. With the communications available in a CV smart corridor environment, these updates can be fed back to the travelers, helping them make environment friendly route choices and encourage environmentally friendly eco driving patterns. The study for this effort was undertaken for a 3-hour PM-peak model run, during which real-time data is used to drive the simulation model. The plots for energy and CO_2 emissions were created dynamically during the run as previously described. The model architecture simulated traffic at 1.3x rate compared to wall clock time and generated dynamic performance evaluation plot at 0.98x rate, for an hour of experiment on a dual processor machine with 192 GB RAM and 2.5 GHz frequency. An example energy heat map generated at the end of the 47th simulation minute is shown in Figure 7, presenting the cumulative energy consumption during the 46-47 simulation minute interval throughout the corridor. The points in the heat map symbolize the midpoint of 200 ft corridor segments. The heat image around each point represents the total energy consumed in the area in the last minute.

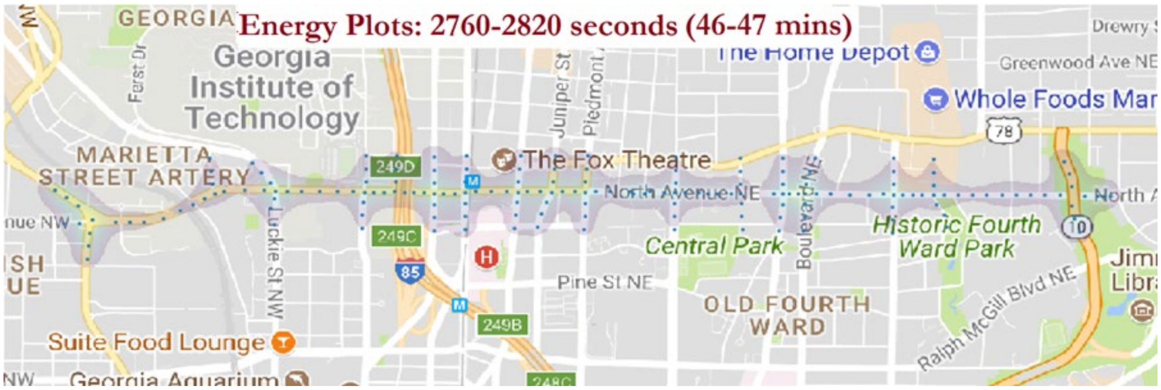


Figure 7: Energy plot generated dynamically during simulation runtime.

3.3.2. Model Sensitivity to Real-Time Input

The model's sensitivity to real-time input is tested by comparing the performance measures for simulation runs with and without real-time input at the intersections of Glen Iris Dr. with North Ave. and Ponce City Market with North Ave., with all other intersections utilizing the calibrated data in both scenarios. For the two scenarios vehicle travel times, energy consumption, and CO_2 emissions performance measures were measured for eight selected routes that traverse the two intersections.

3.3.2.1. Experiment Design

The simulation period for both the real-time model and preset-input model is three hours (10800 sec) with simulation resolution of ten time steps per simulation second. To reflect randomness ten replicate runs were undertaken for each scenario, where the Vissim Random Seed parameter is changed in each replicate. This allowed stochastic variation of vehicle arrivals into the network. Eight routes that go across the real-time driven intersections – Glen Iris Drive and Ponce City Market – are selected to specifically investigate the built model's sensitivity to near-real-time input. Figure 8 shows the selected eight routes, of which the four westbound routes start at the Freedom Parkway intersection and four eastbound routes start near the I75/85 Connector. Additionally, for the eight selected routes, energy consumption and CO_2 emissions with simulation time are compared.

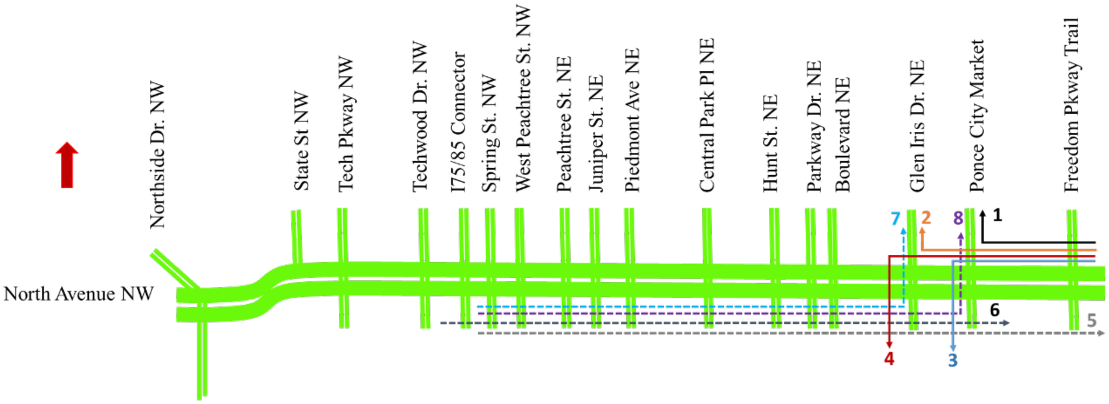
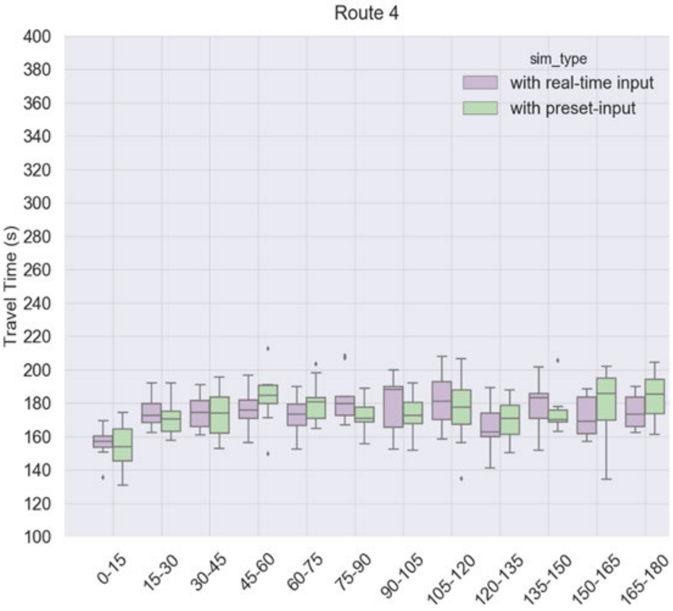


Figure 8: Routes selected for studying model's sensitivity to real-time input.

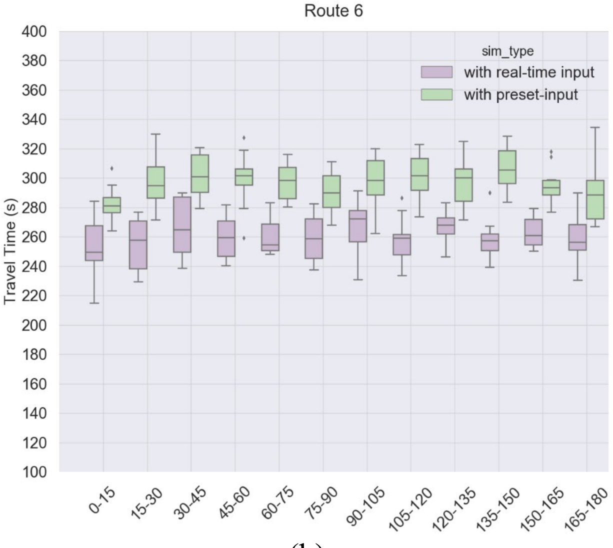
3.3.2.2. Vehicle Travel Time

For brevity, results are given for two routes. Figure 9(a) below shows boxplots of vehicle travel time versus simulation time interval (15 minute intervals) for the ten random runs with real-time input (i.e., driven by field data) and with calibrated data on westbound Route 4. Figure 9(b) shows the same for eastbound Route 6. These box plots show the average vehicle travel times of all vehicles that end their trip on the respective routes, for the corresponding simulation interval on x-axis.

For Route 4, it is observed that the vehicle travel times for the real-time model and the preset-input model are comparable. The two sets of travel times fluctuate similarly over the simulation period. For Route 6, the difference in travel times between real-time and preset-input simulation is larger compared to that observed in Route 4; however, the values are still comparable. The maximum difference in average travel times is of approx. 60 seconds (1 minute) for the simulation interval of 135-150 min. Among the eight selected routes, Route 4 and Route 6 shown in the figures here, recorded the maximum differences in westbound and east bound travel times. The closeness in vehicle travel time values for ten random simulation runs across all routes for the two compared models demonstrates that the results obtained from the real-time input model are within plausible bounds, with the potential that the real-time data is allowing for improved estimations from the simulation. However, field collected travel times are needed to confirm this supposition.



(a)

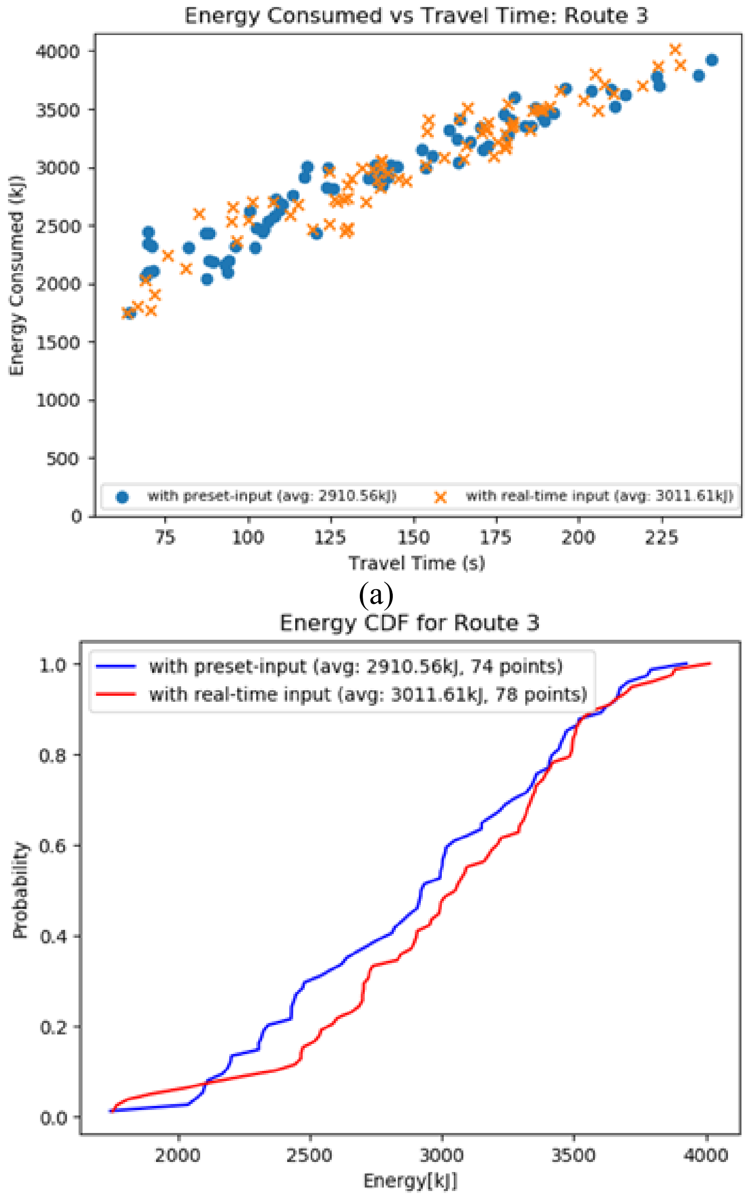


(b)

Figure 9: Average vehicle travel time versus simulation time intervals plots for (a) Westbound Route 4, and (b) Eastbound Route 6.

3.3.2.3. Energy Consumption and CO_2 Emission

The comparison of the energy and emissions performance metrics from one pair of runs is shown in Figure 10. The energy consumption follows a fairly linear pattern with increases in travel time (Figure 10(a)). The longer a vehicle takes to traverse a given path, the more energy it is expected to consume, under most conditions. As reflected in the cumulative density function of energy (Figure 10(b)), the slightly higher travel times in the real-time simulation for Route 3 (whereas in Route 4 and Route 6 above real-time data driven simulation travel times tended to be lower) results in consistently higher energy estimates than the simulation without real-time data. The emissions values followed very similar trends as the energy plots. These results are to be field validated. However, closeness of results for the two scenarios shows the potential of the real-time data driven simulation to provide improved performance metric estimates.



(a)

(b)

Figure 10: (a) Scatterplot for energy consumption versus travel time for Route 3, and (b) Cumulative energy consumption density function for vehicles on Route 3.

3.4. Conclusion and Future Work

This study assesses feasibility aspects of using a real-time data-driven transportation simulation model to evaluate and visualize network performance indices that can be used to provide dynamic feedback to the real world operations environment. A hybrid traffic simulation model, consisting of a mix of pre-programmed and real-time data-driven intersections, is developed. The principal challenge identified in building the hybrid transportation simulation model is in integrating the multiple components underlying the complete model and ensuring they work in synchrony with real-time. However, an architecture was developed to enable such hybrid modeling. The simulation results from the hybrid model were compared with results from the model using preset values. The performance results indicated that with the presented architecture it is plausible to work with the high velocity data while ensuring sufficient responsiveness of the model to input changes.

In the next section the corridor simulation model is expanded to incorporate available real-time data for all intersection in the model. Further, the data streams are investigated for data gaps, and data imputation methods are tested.

(*This effort presented in this chapter has been published in: Saroj, A., S. Roy, R. Fujimoto, A. Guin, and M. Hunter. Smart City Real-Time Data-Driven Transportation Simulation. Proceeding of the Winter Simulation Conference (WSC), Gothenburg, Sweden D.C., Dec 8th - 12th, 2018.)

4. UPDATED MODEL ARCHITECTURE, DATA ISSUES, AND MODEL PERFORMANCE SENSITIVITY TO VOLUME IMPUTATIONS EXPERIMENT

4.1. Introduction

In a smart city, which is equipped with connected infrastructure, traffic data, such as vehicle detections and intersection signal indications are expected to be received in (near) real-time. Utilizing such data, the objective of the near-real-time simulation platform highlighted in this Chapter is the creation of a dynamic data-driven simulation that leverages high-frequency connected data streams to derive meaningful insights about the current traffic state and near-real-time corridor environmental measures. As a first step, a connected-infrastructure data-driven simulation model is developed. The developed model is capable of being driven in near-real-time with high-frequency connected infrastructure traffic volume and signal controller data streams. The model visualizes key traffic and environmental performance measures at near-real-time, providing dynamic feedback to users and transportation stakeholders. Such a data-driven simulation platform can be crucial in providing helpful insights on the effectiveness of new technology deployments. In this context, the built simulation model can be used to test impacts of smart/connected technologies, such as smart signal control or traveler information systems, in real-time.

The developed simulation model emulates traffic on 2.3 miles of the North Avenue Smart Corridor in Atlanta, Georgia. The simulation model architecture feasibility and robustness were initially explored by driving two of the intersections with near real-time data streams and the remaining intersections using preset data, as discussed in Chapter 3. In this initial experiment the overall architecture was found to be capable of inserting the data into the simulation platform, maintaining the faster-than-real-time processing necessary for such a platform to maintain real-time capabilities, and able to provide meaningful traffic operations and vehicle emission estimates along the corridor (Saroj, Roy et al. 2018). For this chapter, the ability to stream the volume and signal data real-time has been expanded to all modeled signalized intersections in the corridor. However, investigation of the data streams themselves has revealed the presence of data gaps. Given the challenges of maintaining data streams in harsh field environments, such data loses should be expected, though such data gaps can impact the performance measure results and insights generated from the model. Thus, the model architecture needs to be enhanced to handle such data losses. Development of an imputation methodology should be informed by an understanding of the impact of data loss, and errors in data imputation, on generated performance measures. This chapter provides an overview of the improved architecture and utilizes sensitivity analysis to explore the impact of the volume data gap imputations on the key performance measures produced by the near-real-time data-driven simulation. Figure 11 shows the 15 intersections of the North Avenue Smart Corridor studied in this experiment.

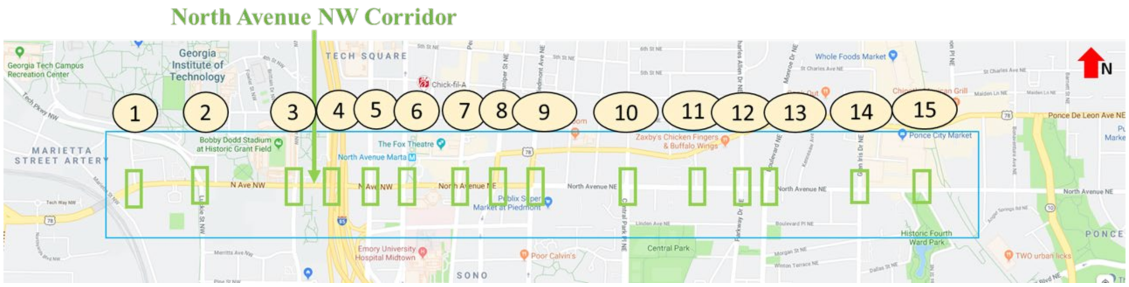
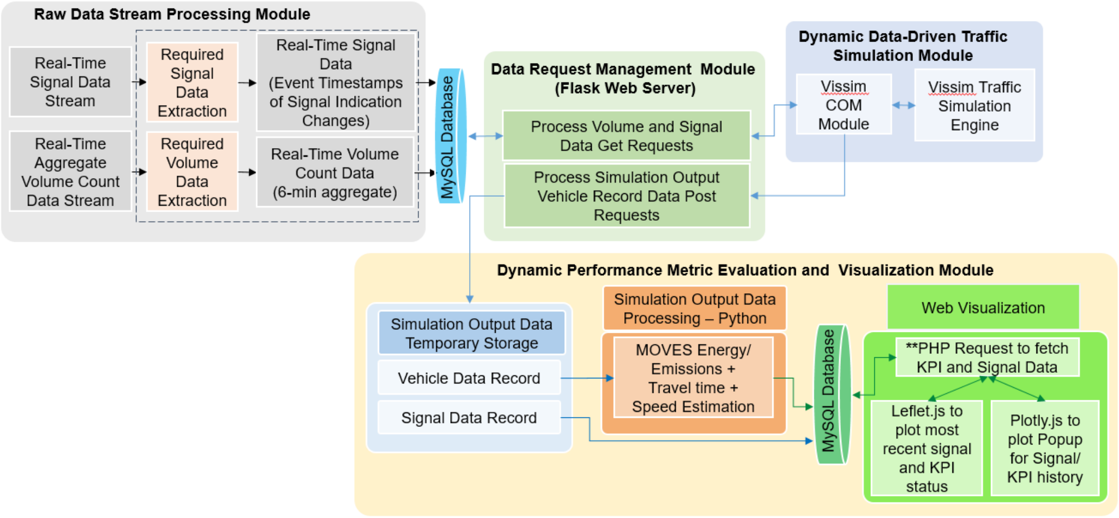


Figure 11: Study corridor-2.3 miles of North Avenue Smart Corridor, including 15 signalized intersections.

4.2. Updated Real-Time Data-Driven Simulation Model Architecture

For simplicity, through the remainder of this chapter “real-time” will be use rather than “near-real-time”. The architecture is currently implemented with a lag of approximately seven minutes due to field data aggregation delays, although it could be implemented with a lag of just a few seconds given a real-time data stream. Figure 12 provides the updated real-time data-driven simulation architecture. The architecture is updated to perform four primary tasks: 1) injection of real-time signal control and volume data streams into traffic simulation model through *raw data stream processing*, 2) model execution through *dynamic data-driven traffic simulation*, 3) *dynamic performance metric evaluation and visualization*, and finally 4) efficient handling of transactions between modules using a *data request management – (Flask Web Server)*. The following sections elaborate on each of these modules.

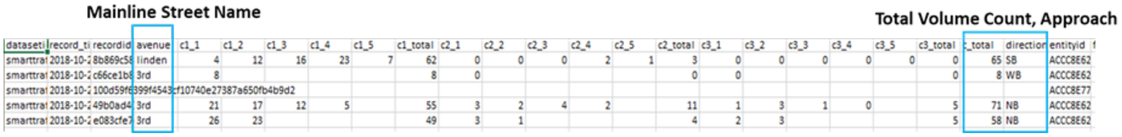

Raw Data Stream Processing Module
Real-Time Signal Data Stream
Required Signal Data Extraction
Real-Time Signal Data (Event Timestamps of Signal Indication Changes)
Real-Time Aggregate Volume Count Data Stream
Required Volume Data Extraction
Real-Time Volume Count Data (6-min aggregate)
MySQL Database
Data Request Management Module (Flask Web Server)
Process Volume and Signal Data Get Requests
Process Simulation Output Vehicle Record Data Post Requests
Dynamic Data-Driven Traffic Simulation Module
Vissim COM Module
Vissim Traffic Simulation Engine
Dynamic Performance Metric Evaluation and Visualization Module
Simulation Output Data Temporary Storage
Vehicle Data Record
Signal Data Record
Simulation Output Data Processing – Python
MOVES Energy/ Emissions + Travel time + Speed Estimation
MySQL Database
Web Visualization
**PHP Request to fetch KPI and Signal Data
Leflet.js to plot most recent signal and KPI status
Plotly.js to plot Popup for Signal/ KPI history

**Note: All components in the architecture run continuous except PHP requests in Web Visualization that are on-demand

Figure 12: Updated real-time data-driven simulation model architecture overview.

4.2.1. Real-Time Raw Data Stream Processing Module

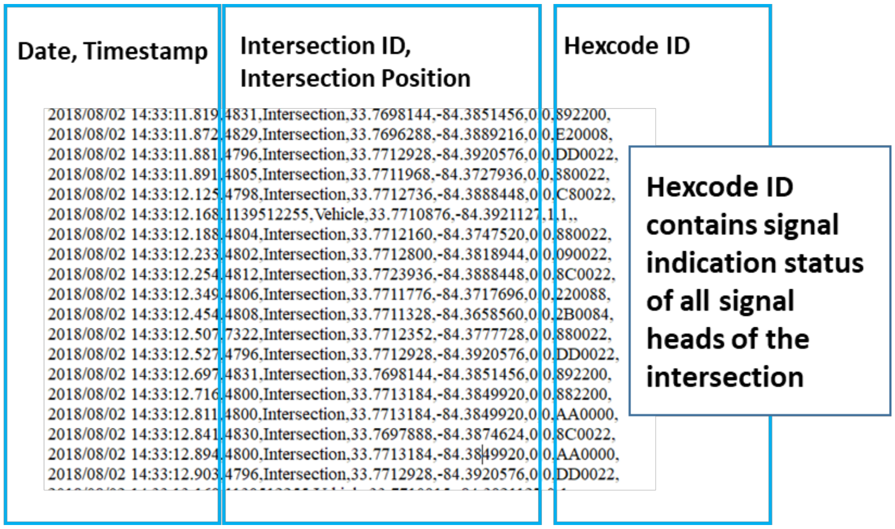
For each intersection, real-time vehicle per-lane counts are collected in the field using video detection and processing. Data are received ("Real-Time Aggregate Volume Count Data Stream", Figure 12) as 6-minute, per-lane aggregate counts. The raw volume data stream (see Figure 13 for a sample) is filtered for the approaches of the 15 studied intersections. Volume count data contains both the timestamp of the start of the interval and the distribution of the volume count per lane, for each of three vehicle classes. The 3 classes are listed as c1, c2, and c3 in Figure 13, where c1_1 refers to the count of vehicle class 1 in lane 1 and similarly c2_5 refers to the count of vehicle class 2 on lane 5. The lane ID (i.e., 1, 2, 3, etc.) follows the serial order from rightmost lane toward the roadway median. In the *raw data stream processing module* lane level raw volume data counts are aggregated across classes and lanes to obtain the approach level by movement aggregate counts, to be used as input for the traffic simulation model.



dataseti	record_ti	recordid	avenue	c1_1	c1_2	c1_3	c1_4	c1_5	c1_total	c2_1	c2_2	c2_3	c2_4	c2_5	c2_total	c3_1	c3_2	c3_3	c3_4	c3_5	c3_total	_total	direction	entityid	f
smarttra	2018-10-2	8b869c58	linden	4	12	16	23	7	62	0	0	0	2	1	3	0	0	0	0	0	0	65	SB	ACCC8E62	
smarttra	2018-10-2	c66ce1b8	3rd	8					8	0					0	0					0	8	WB	ACCC8E62	
smarttra	2018-10-2	100d59f6399f4543cf10740e27387a650fb4b9d2																						ACCC8E77	
smarttra	2018-10-2	49b0ad4	3rd	21	17	12	5		55	3	2	4	2		11	1	3	1	0		5	71	NB	ACCC8E62	
smarttra	2018-10-2	e083cfe7	3rd	26	23				49	3	1				4	2	3				5	58	NB	ACCC8E62	

Figure 13: Snapshot of raw volume count data.

The signal controller data streams are received ("Real-Time Signal Data Stream," Figure 12) at a higher frequency, ranging between 0.1 Hz and 10 Hz. The received data stream contains the signal color indication status of all the signal heads for all intersections. Separate messages are sent for each intersection. Each message contains the state of all signal indications for that intersection, and a message is sent whenever any indication changes at the intersection. However, the raw signal data stream may include repeated redundant messages. For several intersections update message were sent approximately every 2.5 to 5 seconds even when there was no change in indications. From this data stream, messages that reflect a change in the current state of the signal indications (i.e., a light change on any signal head associated with the intersection) are identified to be used as dynamic input to the simulation model. Figure 14 provides a data sample, where each signal message record contains the associated date, timestamp, intersection ID, position of the intersection in latitude/longitude format, and hexcode ID. The hexcode ID is converted to a binary data string that contains signal status of all (max of 8) phases of the intersection at the corresponding timestamp.


Date, Timestamp
Intersection ID,
Intersection Position
Hexcode ID
Hexcode ID contains signal indication status of all signal heads of the intersection

```
2018/08/02 14:33:11.819,4831,Intersection,33.7698144,-84.3851456,0,0,892200,
2018/08/02 14:33:11.872,4829,Intersection,33.7696288,-84.3889216,0,0,E20008,
2018/08/02 14:33:11.881,4796,Intersection,33.7712928,-84.3920576,0,0,DD0022,
2018/08/02 14:33:11.891,4805,Intersection,33.7711968,-84.3727936,0,0,880022,
2018/08/02 14:33:12.125,4798,Intersection,33.7712736,-84.3888448,0,0,C80022,
2018/08/02 14:33:12.168,1139512255,Vehicle,33.7710876,-84.3921127,1,1,,
2018/08/02 14:33:12.188,4804,Intersection,33.7712160,-84.3747520,0,0,880022,
2018/08/02 14:33:12.233,4802,Intersection,33.7712800,-84.3818944,0,0,090022,
2018/08/02 14:33:12.254,4812,Intersection,33.7723936,-84.3888448,0,0,8C0022,
2018/08/02 14:33:12.349,4806,Intersection,33.7711776,-84.3717696,0,0,220088,
2018/08/02 14:33:12.454,4808,Intersection,33.7711328,-84.3658560,0,0,2B0084,
2018/08/02 14:33:12.507,7322,Intersection,33.7712352,-84.3777728,0,0,880022,
2018/08/02 14:33:12.527,4796,Intersection,33.7712928,-84.3920576,0,0,DD0022,
2018/08/02 14:33:12.697,4831,Intersection,33.7698144,-84.3851456,0,0,892200,
2018/08/02 14:33:12.716,4800,Intersection,33.7713184,-84.3849920,0,0,882200,
2018/08/02 14:33:12.811,4800,Intersection,33.7713184,-84.3849920,0,0,AA0000,
2018/08/02 14:33:12.841,4830,Intersection,33.7697888,-84.3874624,0,0,8C0022,
2018/08/02 14:33:12.894,4800,Intersection,33.7713184,-84.3849920,0,0,AA0000,
2018/08/02 14:33:12.903,4796,Intersection,33.7712928,-84.3920576,0,0,DD0022,
```

Figure 14: Snapshot of raw signal event data.

4.2.2. Dynamic Data-Driven Traffic Simulation Module

The six-minute aggregate vehicle count and high-frequency signal state information are inputs to the traffic simulation model. The volume data provides the input traffic on boundary links. The turn-movement ratios for internal links are enabled to change dynamically in the architecture. Where turn and through movements share a lane, turn-movement splits are set based on historic data. The signal data drives all signal changes in the Vissim model. The traffic simulation model is developed in PTV's Vissim 9.00-08. Vissim's COM module is used to feed real-time data into the simulation model during runtime. Thus, the built arrangement simulates the last interval for which data are obtained from the real-time data stream, i.e., there is an approximately 7-minute lag between the simulation model and the field (i.e.. one interval plus fetching delay, discussed further in section 4.3). The simulation resolution is 10 Hz, although signal change events occur at a 1 Hz rate. The 6-minute volume is randomly entered (shifted Poisson inter-arrival headway distribution) across the 6-minute period. Figure 15 provides an overview of the dynamic data driven simulation initialization logic and Figure 16 provides an overview of the simulation runtime logic.

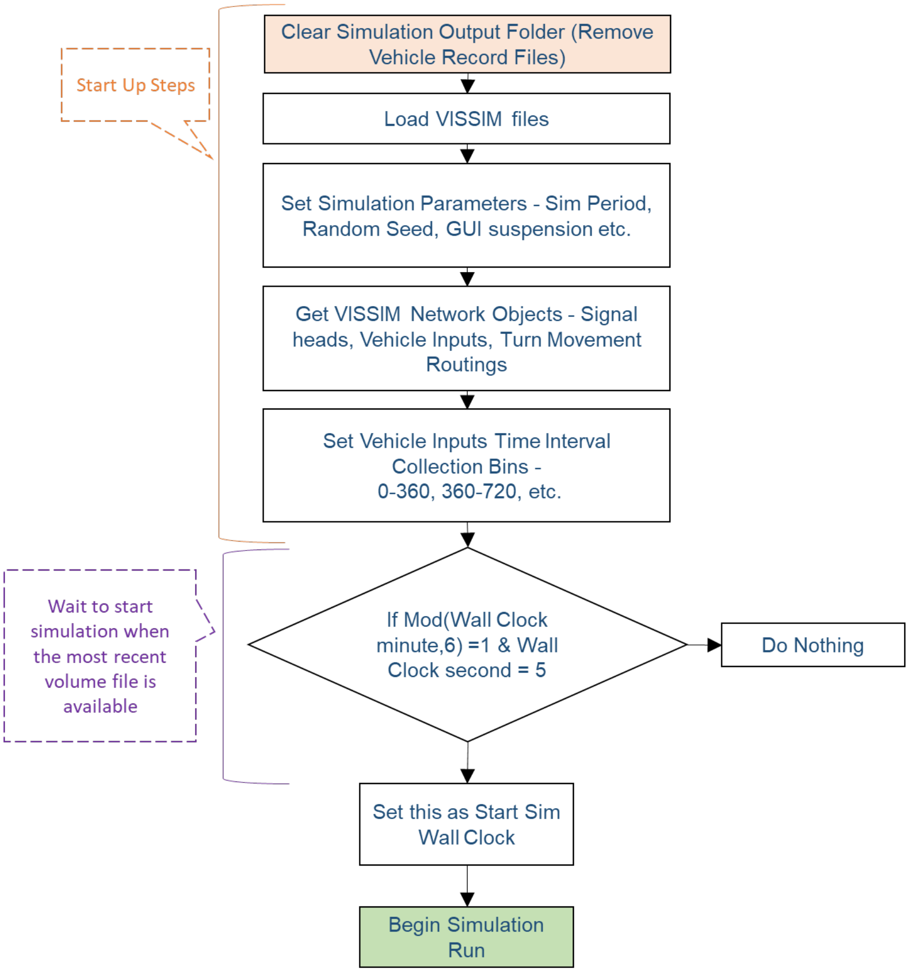

Start Up Steps
Clear Simulation Output Folder (Remove Vehicle Record Files)
Load VISSIM files
Set Simulation Parameters - Sim Period, Random Seed, GUI suspension etc.
Get VISSIM Network Objects - Signal heads, Vehicle Inputs, Turn Movement Routings
Set Vehicle Inputs Time Interval Collection Bins - 0-360, 360-720, etc.
Wait to start simulation when the most recent volume file is available
If Mod(Wall Clock minute,6) =1 & Wall Clock second = 5
Do Nothing
Set this as Start Sim Wall Clock
Begin Simulation Run

Figure 15: Overview of *dynamic data-driven traffic simulation* initialization logic.

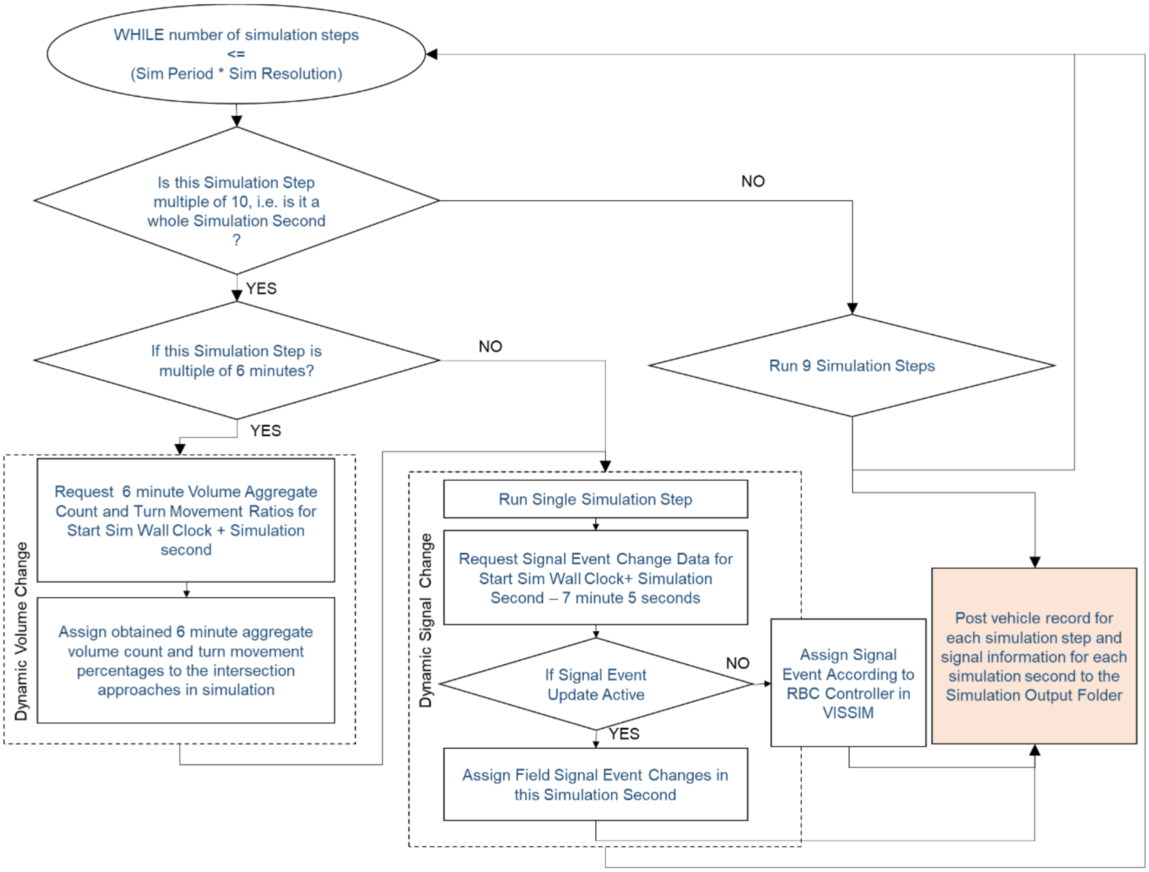

WHILE number of simulation steps
<=
(Sim Period * Sim Resolution)
Is this Simulation Step multiple of 10, i.e. is it a whole Simulation Second ?
NO
YES
If this Simulation Step is multiple of 6 minutes?
NO
YES
Run 9 Simulation Steps
Dynamic Volume Change
Request 6 minute Volume Aggregate Count and Turn Movement Ratios for Start Sim Wall Clock + Simulation second
Assign obtained 6 minute aggregate volume count and turn movement percentages to the intersection approaches in simulation
Dynamic Signal Change
Run Single Simulation Step
Request Signal Event Change Data for Start Sim Wall Clock+ Simulation Second – 7 minute 5 seconds
If Signal Event Update Active
NO
YES
Assign Field Signal Event Changes in this Simulation Second
Assign Signal Event According to RBC Controller in VISSIM
Post vehicle record for each simulation step and signal information for each simulation second to the Simulation Output Folder

Figure 16: Overview of *dynamic data-driven traffic simulation* runtime logic.

4.2.3. Dynamic Performance Measures Visualization Module

The simulation provides a record of position, speed, acceleration, etc. for each vehicle in the network, at a 1 Hz rate. These per-second vehicle record data are used in runtime to estimate performance measures such as travel time on selected routes, queue lengths, energy consumed per vehicle, and emissions generated per vehicle. The estimated performance measures are then visualized on Openstreet maps.

4.2.4. Data Request Transactions Management Module – Flask Web Server

During a run dynamic data requests between the three other modules are handled using Python's Flask web server. A Flask web server is used to generate urls that contain raw data fetched from the *real-time raw data stream processing module* requested by *the dynamic data-driven traffic simulation module*. Further, the flask web server is used to process post requests in the Vissim simulation logic, creating a dynamic log of the simulation output data that contains a record of position, speed, acceleration, etc., for each vehicle in the network. Lastly, the flask web server is also used to facilitate a similar dynamic data request link between the simulation output data log and the *dynamic performance metric visualization* module.

4.3. Software Architecture

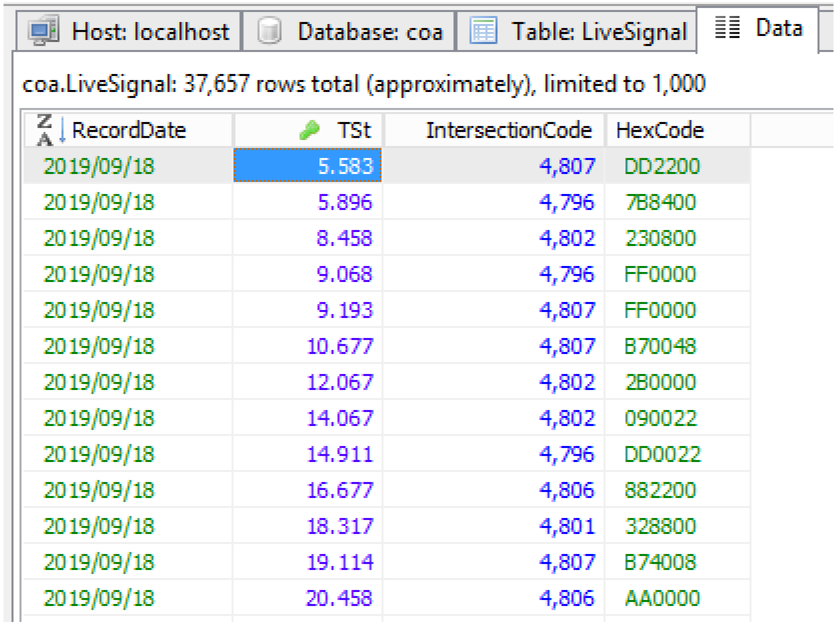
The simulation model architecture is supported by a software architecture that uses different open-source software, programming languages, and open-source modules. The following subsections describe the different components that have been used to create the software architecture, expanding the module discussion in section 4.2.

4.3.1. Extract, Transform, and Load Data

The Extract Transform and Load (ETL) process involves multiple technologies. Several PERL and Python scripts perform the task of continuously fetching data from the sensors in the field and pushing them into a central database repository. The signal database gets updates at a frequency of 10 Hz whereas the volume data is available at a 360 seconds aggregate that is fetched on a 60 seconds update cycle to accommodate possible update latencies at the edge of the detection network.

The central database server has two separate tables for populating live signal and volume data. The first table records change in signal state information as signal state is only transmitted from the field when there is a change for the given intersection. However, several intersections will send messages every 2.5 to 5 seconds, when no state change occurs. Signal states are originally binary state information in the field but are compressed and transmitted in hexadecimal format by the vendor, helping to reduce cellular data usage. A query to the database returns the state in the form of a hexadecimal code as given by the field *HexCode* (sample data may be seen in Figure 17). The python code driving Vissim COM converts this code to binary status information to determine the changes in the signal states at the corresponding intersections. If no *HexCode* is returned, the signal state is left unchanged.

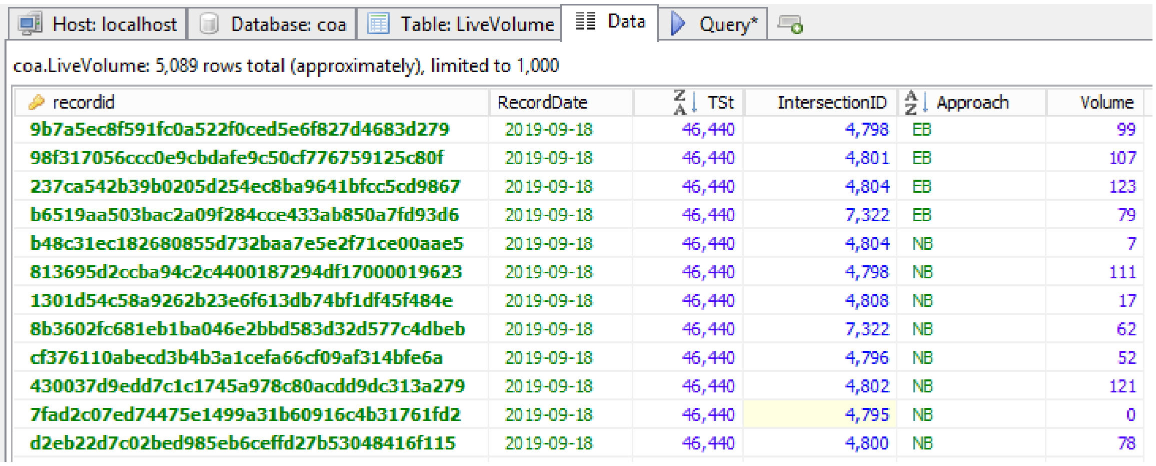
The second table records volumes, where total volume is streamed and recorded every 360 seconds, for each intersection approach (sample data may be seen in Figure 18). The intersection associated with the IntersectionCode in Figure 17 (or IntersectionID in Figure 18) may be found in Table 1. While volume data is available for the intersection at Northside Dr., the signal data is not available. A placeholder identifier of 1234 is used for this intersection. This enables making a query for the intersection at a given simulation time stamp using the same intersection identifier for both signals and volumes. However, in the updated model discussed in this chapter the Northside Dr. intersection is not included as the signal data would need to be imputed for all time periods.



coa.LiveSignal: 37,657 rows total (approximately), limited to 1,000

RecordDate	TSt	IntersectionCode	HexCode
2019/09/18	5.583	4,807	DD2200
2019/09/18	5.896	4,796	7B8400
2019/09/18	8.458	4,802	230800
2019/09/18	9.068	4,796	FF0000
2019/09/18	9.193	4,807	FF0000
2019/09/18	10.677	4,807	B70048
2019/09/18	12.067	4,802	2B0000
2019/09/18	14.067	4,802	090022
2019/09/18	14.911	4,796	DD0022
2019/09/18	16.677	4,806	882200
2019/09/18	18.317	4,801	328800
2019/09/18	19.114	4,807	B74008
2019/09/18	20.458	4,806	AA0000

Figure 17: Sample from signal state data table



coa.LiveVolume: 5,089 rows total (approximately), limited to 1,000

recordid	RecordDate	TSt	IntersectionID	Approach	Volume
9b7a5ec8f591fc0a522f0ced5e6f827d4683d279	2019-09-18	46,440	4,798	EB	99
98f317056ccc0e9cbdafe9c50cf776759125c80f	2019-09-18	46,440	4,801	EB	107
237ca542b39b0205d254ec8ba9641bfcc5cd9867	2019-09-18	46,440	4,804	EB	123
b6519aa503bac2a09f284cce433ab850a7fd93d6	2019-09-18	46,440	7,322	EB	79
b48c31ec182680855d732baa7e5e2f71ce00aae5	2019-09-18	46,440	4,804	NB	7
813695d2ccba94c2c4400187294df17000019623	2019-09-18	46,440	4,798	NB	111
1301d54c58a9262b23e6f613db74bf1df45f484e	2019-09-18	46,440	4,808	NB	17
8b3602fc681eb1ba046e2bbd583d32d577c4dbeb	2019-09-18	46,440	7,322	NB	62
cf376110abecd3b4b3a1cefa66cf09af314bfe6a	2019-09-18	46,440	4,796	NB	52
430037d9edd7c1c1745a978c80acdd9dc313a279	2019-09-18	46,440	4,802	NB	121
7fad2c07ed74475e1499a31b60916c4b31761fd2	2019-09-18	46,440	4,795	NB	0
d2eb22d7c02bed985eb6ceffd27b53048416f115	2019-09-18	46,440	4,800	NB	78

Figure 18: Sample from volume data table

Table 1: Intersection IDs

1234	Northside Dr.
4794	State St. NW
4795	Luckie St. NW
4796	Techwood Dr. NW
4797	I-75/85 Off Ramp
4798	Spring St. NW
4799	West Peachtree St. NW
4800	Peachtree St. NE
4801	Juniper St. NE
4802	Piedmont Ave. NE
7322	Central Park Pl. NE
4804	Hunt St. NE
4805	Parkway Dr. NE
4806	Boulevard NE
4807	Glen Iris Dr. NE
4808	Ponce City Market NE
4809	Freedom Pkwy

For volume, the *recordid* (Figure 18) field is unique and is used as the primary key to populate the live database. This database supports queries with a specified intersection ID, approach, and time stamp. The field TSt denotes the beginning of the 360 second interval for the volume aggregate. For example, the first row of the set of records shown in Figure 18 shows *RecordDate*=2019-09-18, a 6-minute interval start time of TSt=46440 (i.e., seconds from midnight), and an aggregate volume of 99 vehicles for approach 4798 (the code for the EB approach of Spring St.) A query using any timestamp between 46440 and 46800 (ending time for the 360 second interval) for the given date, intersection ID, and approach, will fetch this record.

As the database is updated in real time, it is computationally heavy to maintain an updated index for the tables. To ensure a fast response, it is therefore necessary to limit the size of these buffer tables. At 30 minutes past midnight, the records from the previous day are automatically purged and moved off from both the Volume and Signal tables to archival data tables.

4.3.2. Data Retrieval and Coordination via Flask

With the limitations of the current testbed infrastructure, the volume data aggregate is acquired at 360s aggregate level instead of the ideal per-vehicle-record level. The data is fetched from the field every 60 seconds so there is one minute of additional delay in retrieval of the 360s interval data. Thus, Vissim may be run at an approximately 420 seconds lag from real-time. In this effort a 425 second lag is utilized to minimize impacts due to potential communication delays. The python® script driving the Vissim simulation using the Vissim COM interface, gets the data required to drive the simulation by making data requests to the Flask® server. The Flask server interprets the request and in turn queries the database to fetch the requested signal or volume data and provide it to the Vissim COM script.

4.3.2.1. Flask Requests: Signal

At the start of a simulation driven via the Vissim COM script, the epoch time (a time base common in computer programming) for the start of the simulation is recorded. Since Flask does not store any data, the starting epoch is passed as an argument for every query throughout the run-time of the simulation. The COM script makes a GET request for all signal changes at the start of every simulation second for the next second. The signal data request is made via a REST API call using a URL formatted as follows:

> http://ip_address:port_number/LiveSignal/startepoch=start_epoch&simtime=simulation_time

For example, a simulation, started at an epoch time of 1568660943 seconds, asking for signals for simulation time t=40s, with Flask running at IP address 100.100.100.101 and port 5000, will make a GET request with the URL:

> http://100.100.100.101:5000/LiveSignal/startepoch=1568660943&simtime=40

The URL request triggers a python function that takes start epoch time and simulation time as arguments and pulls relevant signal change events for the particular second from the signal table in the database. For the current testbed network, Vissim simulation rate typically varies between 0.8x to 1.2x, depending dynamic simulation parameters such as traffic demand (i.e., the number of vehicles being simulated), real-time signal data availability (i.e., the number of signal changes executed at a simulation second), etc.

For the current testbed network, Vissim is typically capable of running the simulation faster than real-time. Where Vissim is able to run faster that real-time, to ensure a Vissim run is made synchronously with lagged real-time (real time - 425s), the signal query is held back by the Flask server and executed only when the wall clock reaches the real-time equivalent of the second that is being simulated. For example, if a signal GET request for simulation time *st*=40s for a simulation started at epoch of *se*=1568660943s is made when the current lagged wall time *e* is 1568660982.5s, the system will wait for *se+st-e*= 0.5s, before making the query. This ensures that the data is actually available in the database before the query is executed (barring data transmission failures from the field). Where the simulation runs at slower than real-time rate, the volume query is made for *se+st*, and signal query is made for *se+st-425*.

To accommodate field data losses, this function also checks if a particular signal does not show any changes in the last 4 minutes. In such a situation, it sends an indication to the COM script, in response to which the COM script switches that signal over to be driven by Vissim RBC (Ring-Barrier Control, a common signal control method in an offline Vissim model) until data is again available for the intersection.

4.3.2.2. Flask Requests: Volume

Every 360 seconds, Vissim COM makes a GET request for volume at every entry-point of the network, for the next available volume bin. As stated there is currently a 425 second lag in the simulation and wall clock time (i.e., the current time). However, this lag is a function of the data availability and a smaller volume bin size and lag could be implemented within the framework.

The URL for the requests uses the following format:

> http://ip_address:port_number/LiveVolume/startepoch=start_epoch&intersection=intersection_code&approach=approach&simtime=simulation_time

For example, a simulation, started at an epoch time of 1568660943 seconds, asking for volume for northbound (NB) at State Street (denoted by code 4794), for simulation time interval t=360s-720s, with Flask running at IP address 100.100.100.101 and port 5000, will make a GET request with the URL:

> http://100.100.100.101:5000/LiveSignal/startepoch=1568660943&intersection=4794&approach=NB&simtime=360

Not unlike the URL for signal data, a function is triggered by this URL to make the correct query from the volume table in the central database. Since the signal queries already force a synchronization of Vissim simulation clock to real time at the beginning of every simulation second, the volume queries do not have to enforce the real-time synchronization functionality.

4.3.3. Real-time Trajectory and Signal Data-processing

Vissim COM makes a POST request using Python flask to export the trajectory data. The data is packaged in JSON format for standardization and ease of consumption and is broadcasted at a 1 Hz frequency. The trajectory data includes relevant information about every vehicle in the network such as coordinates, speed, acceleration, headway, etc., at a 10 Hz level. The data is written out to text files as well as to a buffer table in the database. The trajectory data exported by Vissim is then parsed in real-time by a python script for the next step of data processing and KPI calculation. The python script is triggered whenever the POST data is received.

In addition, recall that the signal states are not guaranteed to be available from the field devices, potentially due to device, communication, or other failures. When the data is not available, Vissim is allowed to fall back to operation via the default RBC. However, in these instances the state of the signal is not available in the input stream. To ensure that the status of the signal can be shown in the real-time web interface, it is necessary to export the data from Vissim as well as the input stream. Vissim COM makes additional flask POST calls to export/write the traffic signal status data at the end of every simulation second. The data received is parsed by a python script and pushed into the signal status table in the database for use by the visualization interface.

4.3.4. Trajectory Data Condensation

The trajectory data as parsed from the previous step has information for every vehicle in the network every tenth of a second. To reduce the noise in the data, especially for acceleration values, every 10 records of each vehicle are condensed into a single record by taking the median of the 10 records. The resultant 1 Hz condensed trajectory data is then processed using the MOVES matrix based computation methodology to evaluate KPIs; including, but not limited to energy consumption and vehicular emission aggregates of the network. To study the propagation of KPIs through the network with changing volume and signals, the entire network was subdivided into smaller 200 ft. segments. The KPI values are aggregated from the individual vehicle trajectory points within the segment and reported for the entire network. This reporting occurs with minimal delay, thus posted KPIs are at an approximately 425s delay, resulting from input data delays as previously discussed.

4.3.4.1. Demarcations of 200 ft. Segments

As stated, to ensure uniformity and standardization of reporting, the KPIs were reported as aggregates over standard 200 ft. lengths of roadway units or "segments" that were quantitatively similar, although not exactly the same due to roadway geometry constraints. Barring a few exceptions, the distance between two adjacent intersections in the study network was always higher than 400 ft. The following rules were used to ensure consistency in the demarcations of the segments. Figure 19 shows an example division of a two-intersection corridor.

(1) Each intersection box was separated out as individual segment, irrespective of length. This allowed for the intersections to be treated separately.

(2) The segmentation between intersections started at each intersection and progressed toward the center between the intersections. The middle segment may have a length other than 200ft, although will always be less than 400 feet. Where the middle section would be less than 200 feet this distance would be divided between the two adjacent segments, insuring a minimum segment length of 200 feet.

(3) Within the network, segments are unidirectional. Therefore, each direction of traffic has a separate segment ID.

(4) Within the intersection proper, turn movements require the assignment of vehicle data to two segments. For this effort, the trajectory points lying inside an intersection "box", are associated with the segment with the most similar bearing, as will be seen in the next section.

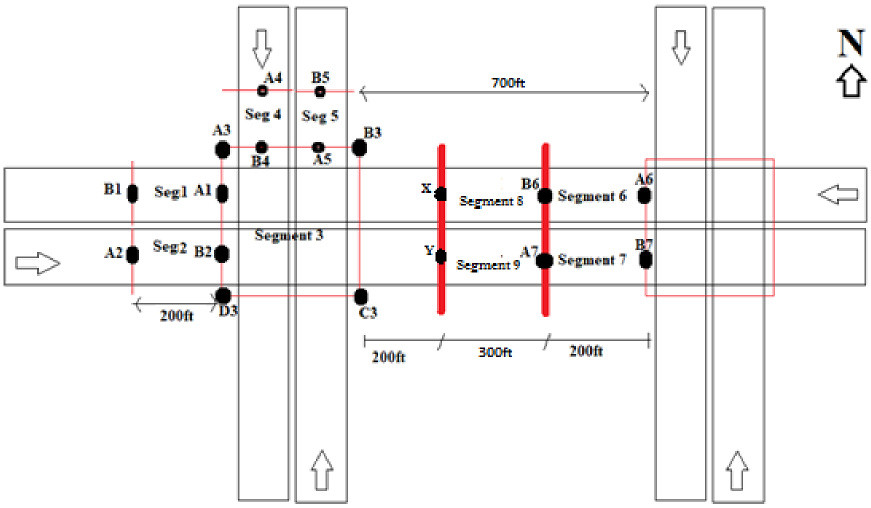


Figure 19: Demarcation process for segments

4.3.4.2. Associating coordinates to segments

Vehicle trajectory data from Vissim includes front and rear coordinates of each vehicle. These coordinates are used to calculate the heading of each vehicle. Using the calculated heading and the x, y coordinate of a vehicle, a vehicle is assigned to a particular segment of the network using coordinate geometry. The steps involved are as follows. To begin with, a rectangular boundary is created around the intersection segments. The vehicles lying inside an intersection boundary, as determined by comparison of the coordinates of the vehicles and the corners of the rectangle around the intersection, are set aside to be treated with step (IV). The remaining data-points are outside the intersection bounds and are processed using steps (I) to (III).

Step (I) -Match Vissim® vehicle heading to segment direction

The vehicle heading extracted from Vissim® trajectory files is used to determine the directionality of the segment to which the vehicle is to be assigned. A threshold of 30 degrees of deviation between the vehicle heading and the segment heading is assumed to be acceptable for matching purposes. Therefore if a vehicle heading is within 30 degrees of the segment heading, it is assumed to be aligned with that segment. The segment with traffic in the opposite direction is likely to have a difference of 180±30º from the vehicle heading. Thus, a necessary but not sufficient condition for a Vissim vehicle to be associated to a segment, the heading vector of a vehicle should not make more than a 30 degree offset from the direction vector of the 200 ft. segment. As this is a simple first order calculation, the computation time of this calculation is low. Hence, this is used as the first step in reducing the number of segments to which a vehicle location can be plausibly associated, e.g. vehicles are divided into those headed northbound, southbound, eastbound, and westbound.

Step (II): Width buffer threshold for association of a data-point to a given segment

A segment is defined by the starting and ending x,y coordinates. It is a linear definition as seen in Figure 20. To take into account the roadway width, a buffer is taken on either side of the segment. With three lanes assumed for each direction of the roadway, and allowing for the curbs, a buffer of {3*13ft (lane width) + 10ft (for curbs)} results in approximately 50 feet per roadway direction. This is further multiplied by 2 to allow for error in x,y placement and to allow for roadway curvature. Thus, the entire buffer utilized for a Vissim® segment is 200ft, i.e. 100ft on each side of the roadway.

Thus, for the specific calculations, if the coordinates of the ends of a segment are (X_1, Y_1) and (X_2, Y_2) respectively, the line joining them "*l*" is given by the equation in the x-y plane:

l: $(y - Y_1) = s*(x - X_1)$ …………..…………..…………..…………..………….. (1)

Where the slope is calculated as:

$s = (Y_2 - Y_1)/(X_2 - X_1)$ …………..…………..…………..…………..………….. (2)

The condition that the coordinate (X, Y) of the data-point should lie within the 100 ft on either side of the centerline is given by:

| $(Y-Y_1) – s*(X-X_1)|/ \sqrt{}\,(s^2 +1) < 100$ ………..…………..………..………….. .. (3)

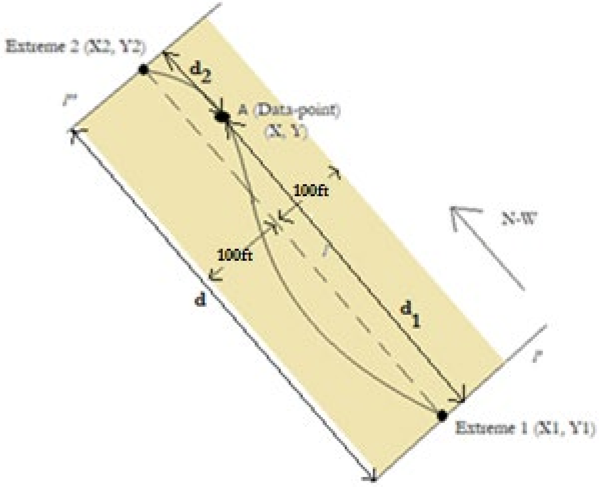


Figure 20: Association of data points to a segment

Step (III): Length buffer threshold for association of a data-point to a given segment

As depicted in Figure 20, the data-point, to be associated with a segment, has to have its coordinates inside the shaded area. In the Step 2 check, the point has been verified to lie within 100 ft of the center line of the segment. The current check ensures that the data-point lies between the end nodes in the segment as well. Perpendicular projections to *l* drawn at each of the extreme nodes (*l'* and *l''*) are given by the following equations:

l': $(x - X_1) + s* (y - Y_1) = 0$ -- (4)

$l'': (x – X_2) + s^* (y – Y_2) = 0$ -- (5)

For the vehicle to be within the bounds in Figure 20, the distance of the vehicle (x, y) from each extremity, as denoted by *l'* and *l''* respectively, will be less than the Euclidean distance between the extremities. These conditions are given by the following equations:

$d_1 = |(X - X_1) + s^* (Y - Y_1)|/ \sqrt{} (s^2 +1) <d$ -- (6)

$d_2 = |(X – X_2) + s^* (Y – Y_2)|/ \sqrt{} (s^2 +1) <d$ -- (7)

Where d = linear distance between (X_1, Y_1) and (X_2, Y_2)

$= \sqrt{}\{(X_2 - X_1)^2 + (Y_2 - Y_1)^2 \}$ -- (8)

Step (IV): Associate data-points in an intersection segment

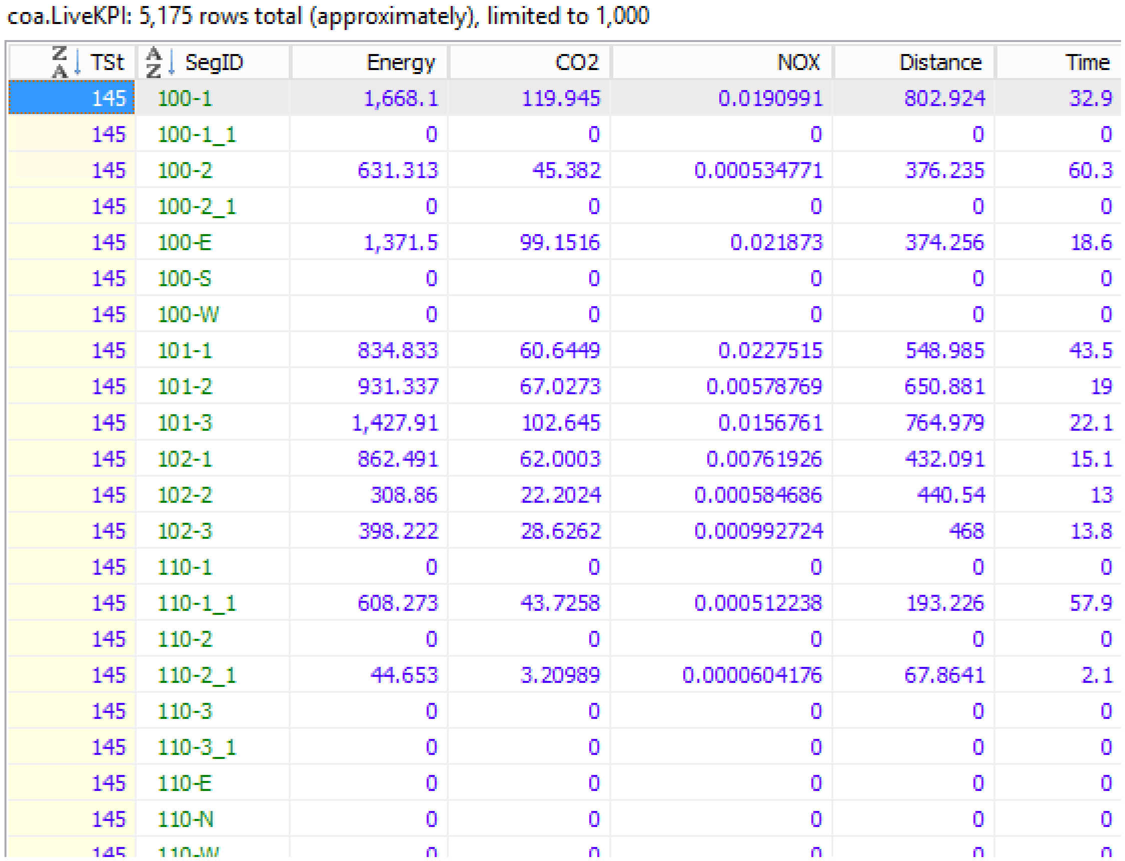
For a data-point within the intersection bounds, the heading of the data-point is used to assign the data-point a direction (E, W, N, or S). For example, for a point that lies inside the box that is demarcated as intersection segment '70', if the heading is between -45° and +45°, the point is assigned North and the data-point is associated with segment '70-N'. A specified association of data-points to a particular directional segment within the intersection enables the study of the transition and handover of the KPI values from one segment to another through the intersection.

4.3.5. KPI calculation

Using the MOVES-matrix (Guensler, Liu et al. 2017), estimates of vehicular emission and energy consumption are computed using the speed, acceleration, and grade associated with each vehicle trajectory data-point. Along with emission and energy, the distance and timestamp associated with the data-point are added to the running total for the respective variable for each segment. The distance and time parameters are used to compute the time-mean speed associated with each segment. The computed time-mean speeds are then used to compute the average travel time in each segment. The KPI aggregates are computed every 10 simulation seconds and populate the KPI table in the database. The table contains all the KPI values along with the aggregate distance, and the aggregate travel time for a given time-stamp interval. Figure 21 provides an example of the KPI table. At any given point of time, the KPI values only for the last 360s interval are stored in the buffer table for retrieval by the visualization interface. The KPIs visualized for this network with the KPI table are:

- Energy
- CO_2 emissions
- NOX emissions
- Speed
- Travel time

The methodology for these calculations may be found in (Liu, Xu et al. 2017).



coa.LiveKPI: 5,175 rows total (approximately), limited to 1,000

TSt	SegID	Energy	CO2	NOX	Distance	Time
145	100-1	1,668.1	119.945	0.0190991	802.924	32.9
145	100-1_1	0	0	0	0	0
145	100-2	631.313	45.382	0.000534771	376.235	60.3
145	100-2_1	0	0	0	0	0
145	100-E	1,371.5	99.1516	0.021873	374.256	18.6
145	100-S	0	0	0	0	0
145	100-W	0	0	0	0	0
145	101-1	834.833	60.6449	0.0227515	548.985	43.5
145	101-2	931.337	67.0273	0.00578769	650.881	19
145	101-3	1,427.91	102.645	0.0156761	764.979	22.1
145	102-1	862.491	62.0003	0.00761926	432.091	15.1
145	102-2	308.86	22.2024	0.000584686	440.54	13
145	102-3	398.222	28.6262	0.000992724	468	13.8
145	110-1	0	0	0	0	0
145	110-1_1	608.273	43.7258	0.000512238	193.226	57.9
145	110-2	0	0	0	0	0
145	110-2_1	44.653	3.20989	0.0000604176	67.8641	2.1
145	110-3	0	0	0	0	0
145	110-3_1	0	0	0	0	0
145	110-E	0	0	0	0	0
145	110-N	0	0	0	0	0
145	110-W	0	0	0	0	0

Figure 21: Sample KPI visualization lookup-table

For web-visualization, the *leaflet.js* module (Agafonkin 2019), a JavaScript module that uses *OpenStreetMaps* (OpenStreetMaps n.d.), is used to create several of the map overlay functionalities. PHP calls to the central database are used for retrieving the real-time KPI data generated from the Vissim data as described in the previous section. As the KPI values are generated every 10 seconds these PHP calls are made at a 0.1 Hz frequency.

In addition to the KPIs, the web interface also displays the traffic signal states for the intersections in the network. Similar to the KPI values, only the last 360s interval of signal data for the signal-heads is retained in the buffer tables at any point in time. For updating the signal data, the PHP queries are made at a 1 Hz rate to enable faster refresh rates for the visualization of the signals.

Alongside the instantaneous visualization, popup charts are used to display a brief history of the KPIs and signal charts. Popup charts are displayed when a segment or signal icon is clicked. The charts show the last 360 seconds of a KPI value or signal state, as appropriate. The *Plotly* module (Plotly 2019) is used to generate these charts. The signal status charts update every 1 second and the KPI value charts refresh every 10 seconds, in congruence with the refresh rates of the instantaneous values in the map.

4.3.1. Visualization frontend

Two versions of the visualization front end are currently deployed and operational in parallel. The real-time version [http://realtime.ce.gatech.edu/coa/now] operates with an implementation of the full architecture described in the preceding sections. There is also a pseudo-real-time version [http://realtime.ce.gatech.edu/coa/] that is operated as a stable demonstration. The pseudo-real-time version is purely for demonstration purposes in the eventuality that the real-time version is unstable (which is expected since it is continuously under development, fed by active research).

To create the pseudo-real-time demonstration version, the website bypasses running Vissim in real-time and uses archival pre-computed KPI and signal status data for March 18, 2019. The archival data is used to populate the 360s buffer tables for the KPI and the status by synchronizing the March 18th data to the current local time (EDT). Figure 22 is a screenshot of the website that visualizes carbon dioxide emissions for the corridor on March 18, 2019. The popups shown in the figure are for historical emissions for the vehicles going WB at Spring St. and signal states experienced by the WB approach for the last 360 seconds. The charts are expected to be intuitively simple to interpret. For example, once the signal turns RED (as seen in the lower chart in Figure 22) all vehicles stop and the emissions are minimal (as seen in the top chart in Figure 22). Subsequently, when the vehicles accelerate from stationary state, the net emissions are observed to increase.

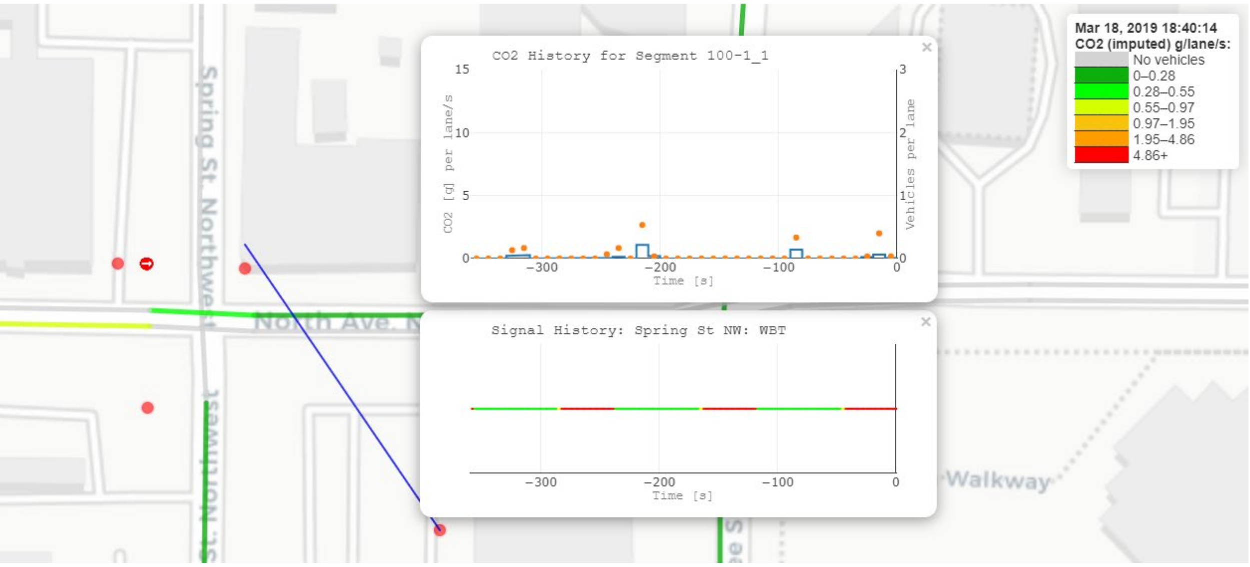


Figure 22: Screenshot of visualization website for March 18, 2019

4.4. Investigation of Real-Time Data Streams

While the model architecture has been shown to be feasible, an investigation of the real-time data streams used as input to the simulation revealed data loss in the data streams. Such missing data, if left absent or incorrectly imputed, is likely to impact the simulation results. Investigation of the streams revealed data gaps are likely related to communications (e.g., dropped or highly latent messages), equipment failure, or data message processing.

The volume and signal data were transmitted through separate systems, albeit both cellular-based; thus, data loss was not correlated between the two data streams. The key characteristics of the observed data loss events are presented in the next sections.

4.4.1. Volume Data Streams

The raw real-time volume data streams contain information such as the timestamp of the start of the volume interval, the volume count distribution per vehicle class per lane, speed, occupancy, etc., for each record, where a record is created for each intersection approach. To study data gaps in the volume data stream, the raw data is reduced to contain only the required information for each record and processed to a dataset that clearly identifies the presence and absence of volume data for each interval of the day, i.e., identifies data gaps.

4.4.1.1. Raw Volume Data Processing to Identify Missing Volume Intervals

Using the four-step filter process shown in Figure 23, a final *Standardized Name Table* is developed from the database table containing the raw volume data allowing for a standardized input into the simulation model. This table filters unused data, converts all roadway naming and lane assignments to a standardized, consistent format, and modifies the timestamps (converting from Greenwich Mean Time (GMT) to Eastern Standard Time (EST)). In generating the *Standardized Name Table,* Lane ID modification was undertaken for the approaches where incomplete or incorrect lane IDs were identified. For example, if an approach with three lanes had data for only 2 lanes, listed as c1_1 and c1_2, it was assumed that the detections were present on the through lanes. With this assumption and according to the original lane ID numbering scheme (lane ID numbering starts from right most lane), the IDs for these lanes would be c1_2 and c1_3, with c1_1 being the missing detection ID name. For consistency in identification of missing detections the names are modified to match the original ID numbering scheme.

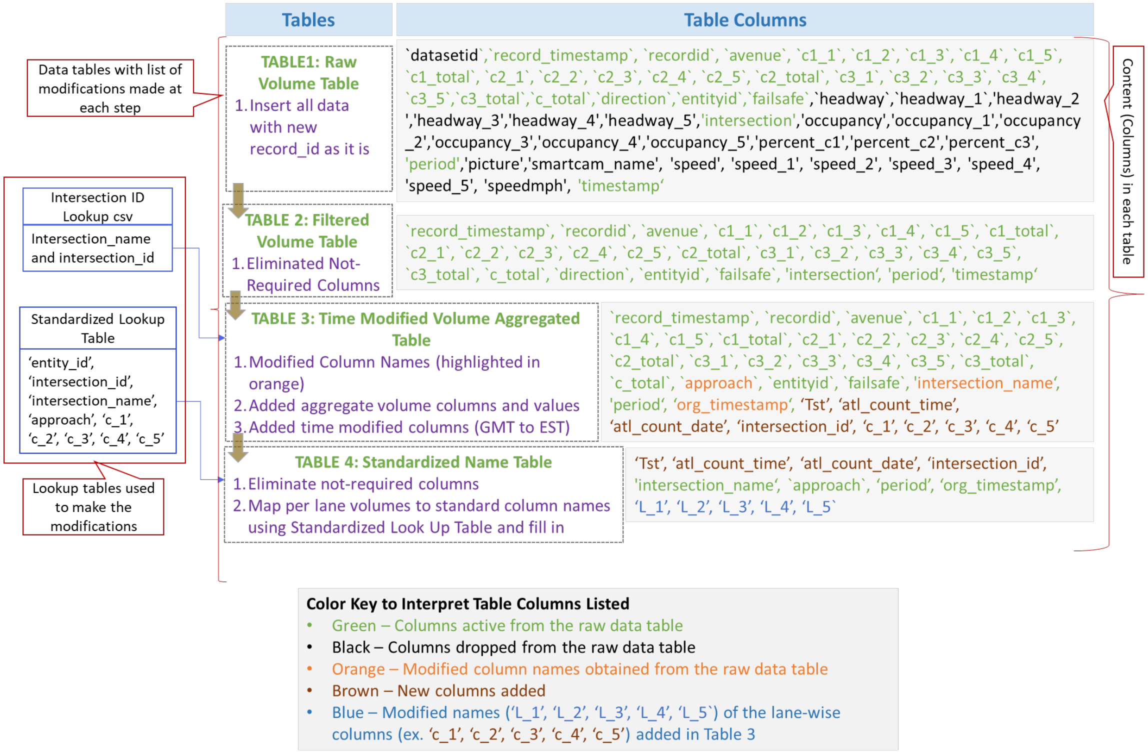
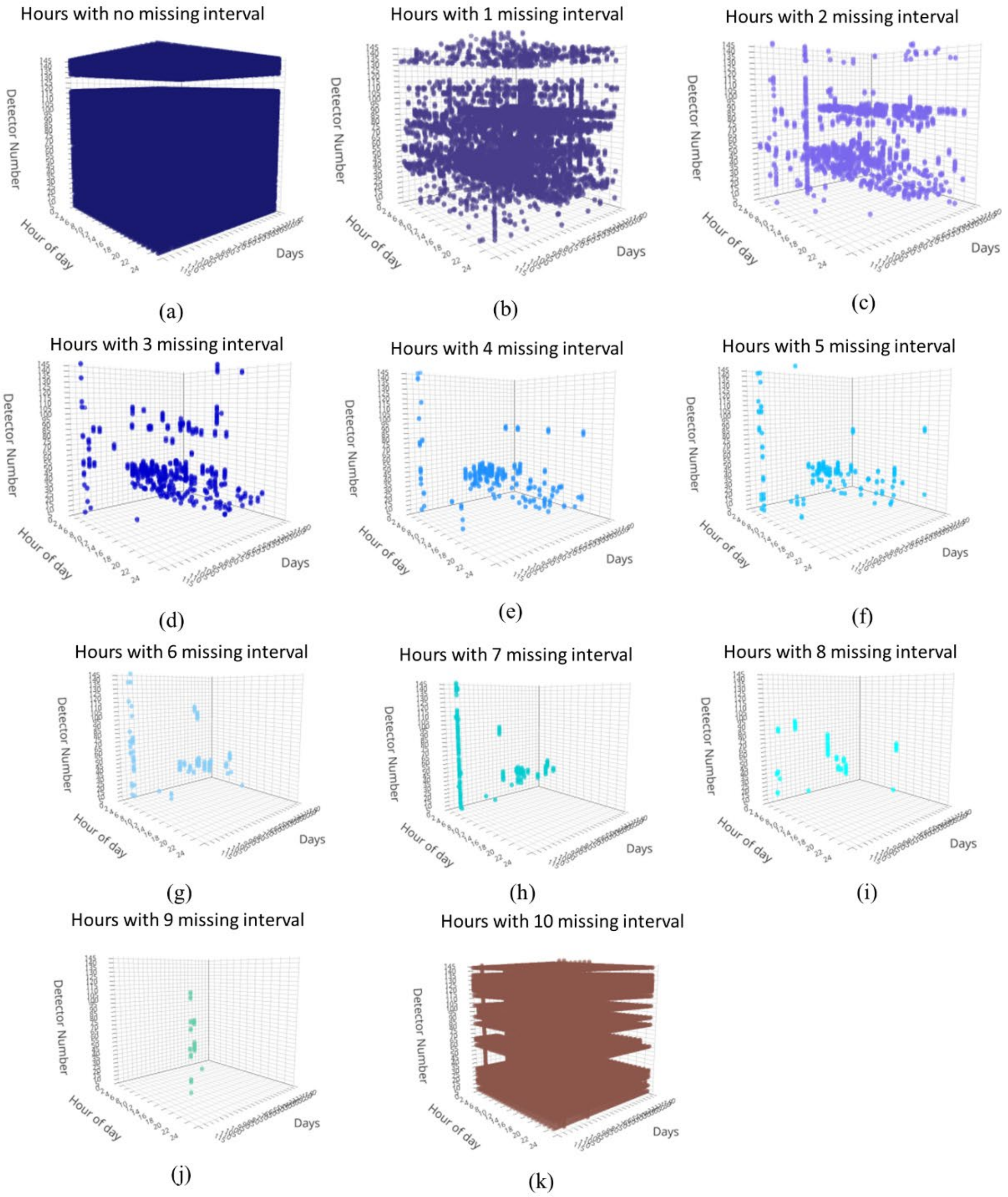

Tables
Table Columns
Data tables with list of modifications made at each step
TABLE1: Raw Volume Table
1. Insert all data with new record_id as it is
`datasetid`,`record_timestamp`, `recordid`, `avenue`, `c1_1`, `c1_2`, `c1_3`, `c1_4`, `c1_5`, `c1_total`, `c2_1`, `c2_2`, `c2_3`, `c2_4`, `c2_5`, `c2_total`, `c3_1`, `c3_2`, `c3_3`, `c3_4`, `c3_5`,`c3_total`,`c_total`,`direction`,`entityid`,`failsafe`,`headway`,`headway_1`,'headway_2','headway_3','headway_4','headway_5','intersection','occupancy','occupancy_1','occupancy_2','occupancy_3','occupancy_4','occupancy_5','percent_c1','percent_c2','percent_c3', 'period','picture','smartcam_name', 'speed', 'speed_1', 'speed_2', 'speed_3', 'speed_4', 'speed_5', 'speedmph', 'timestamp'
Content (Columns) in each table
Intersection ID Lookup csv
Intersection_name and intersection_id
TABLE 2: Filtered Volume Table
1. Eliminated Not-Required Columns
`record_timestamp`, `recordid`, `avenue`, `c1_1`, `c1_2`, `c1_3`, `c1_4`, `c1_5`, `c1_total`, `c2_1`, `c2_2`, `c2_3`, `c2_4`, `c2_5`, `c2_total`, `c3_1`, `c3_2`, `c3_3`, `c3_4`, `c3_5`, `c3_total`, `c_total`, `direction`, `entityid`, `failsafe`, 'intersection', 'period', 'timestamp'
Standardized Lookup Table
'entity_id', 'intersection_id', 'intersection_name', 'approach', 'c_1', 'c_2', 'c_3', 'c_4', 'c_5'
TABLE 3: Time Modified Volume Aggregated Table
1. Modified Column Names (highlighted in orange)
2. Added aggregate volume columns and values
3. Added time modified columns (GMT to EST)
`record_timestamp`, `recordid`, `avenue`, `c1_1`, `c1_2`, `c1_3`, `c1_4`, `c1_5`, `c1_total`, `c2_1`, `c2_2`, `c2_3`, `c2_4`, `c2_5`, `c2_total`, `c3_1`, `c3_2`, `c3_3`, `c3_4`, `c3_5`, `c3_total`, `c_total`, `approach`, `entityid`, `failsafe`, 'intersection_name', 'period', 'org_timestamp', 'Tst', 'atl_count_time', 'atl_count_date', 'intersection_id', 'c_1', 'c_2', 'c_3', 'c_4', 'c_5'
Lookup tables used to make the modifications
TABLE 4: Standardized Name Table
1. Eliminate not-required columns
2. Map per lane volumes to standard column names using Standardized Look Up Table and fill in
'Tst', 'atl_count_time', 'atl_count_date', 'intersection_id', 'intersection_name', `approach`, 'period', 'org_timestamp', 'L_1', 'L_2', 'L_3', 'L_4', 'L_5`
Color Key to Interpret Table Columns Listed
Green – Columns active from the raw data table
Black – Columns dropped from the raw data table
Orange – Modified column names obtained from the raw data table
Brown – New columns added
Blue – Modified names ('L_1', 'L_2', 'L_3', 'L_4', 'L_5`) of the lane-wise columns (ex. 'c_1', 'c_2', 'c_3', 'c_4', 'c_5') added in Table 3

Figure 23: Four-steps of data processing to obtain the *Standardized Name Table* from *Raw Volume Table*

4.4.1.2. Volume Data Gaps Analysis Aided with Interactive 2D and 3D Visualizations

Analysis of volume data streams received for 112 days across the months of February, March, April, and May 2019 revealed gaps in the volume data. On plotting the missing data intervals on 3-D and 2-D dynamic visualizations and heat maps, different missing volume patterns were observed. Figure 24 and Figure 25 show 3D and 2D representations of missing volume patterns over all days. The 3D plots, Figure 24, show the hours with (a) no missing data, (b) one missing six-minute interval, (c) two-missing 6-minute intervals, etc. It is seen that there is a significant level of missing data, with the most frequently occurring being one missing interval or all intervals missing within an hour. Figure 25 shows the aggregation of data loss over 28 day periods. Within this figure it can be seem that data loss is not a completely random process, with outages more likely across certain detectors and times of day.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 24: 3D visualizations of missing data intervals by hour, with 24-hours of a day on the x-axis, 112 days on the y-axis, and detectors on the z-axis. Figures (a-k) show hours with 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, or 10 6-minute missing intervals.

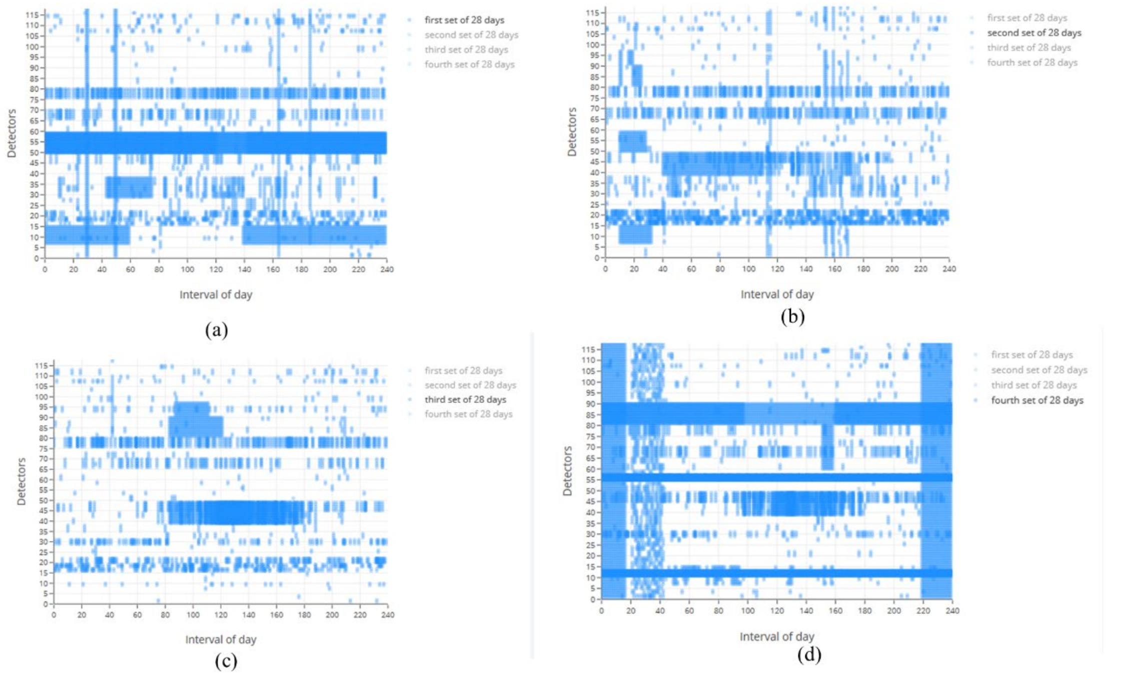


Figure 25: Interactive 2D visualizations showing intermittent missing data patterns, aggregated over days: (a) 1-28, (b) 29-56, (c) 56-84, and (d) 85-112.

Figure 26 is a heat map showing the presence (in light blue) and absence (in dark blue) of data availability for a typical day. The x-axis is divided into 240, 6-minute intervals (i.e., the number of 6-minute intervals in a 24-hour period), from midnight-to-midnight. The y-axis represents 147 detectors spread across the 15 intersections included in the model. While at some detectors no data was obtained for the 112 days, implying complete data loss, at other detectors intermittent data loss with varying patterns from day-to-day was observed.

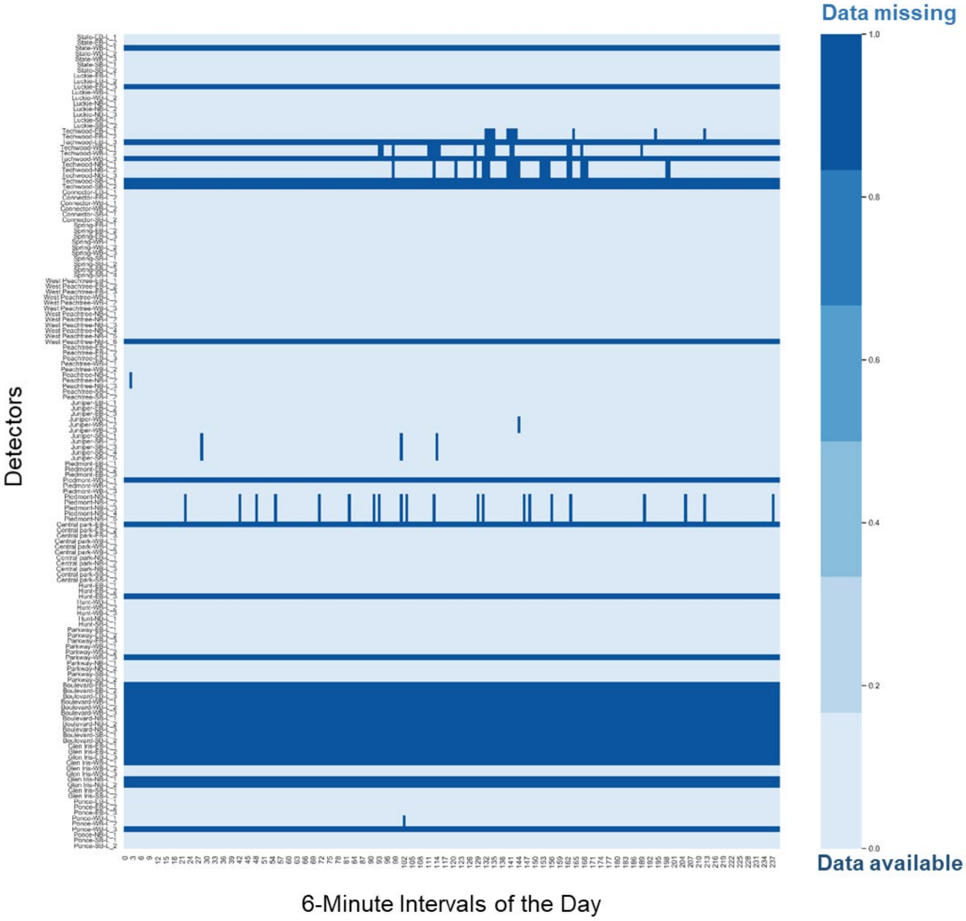
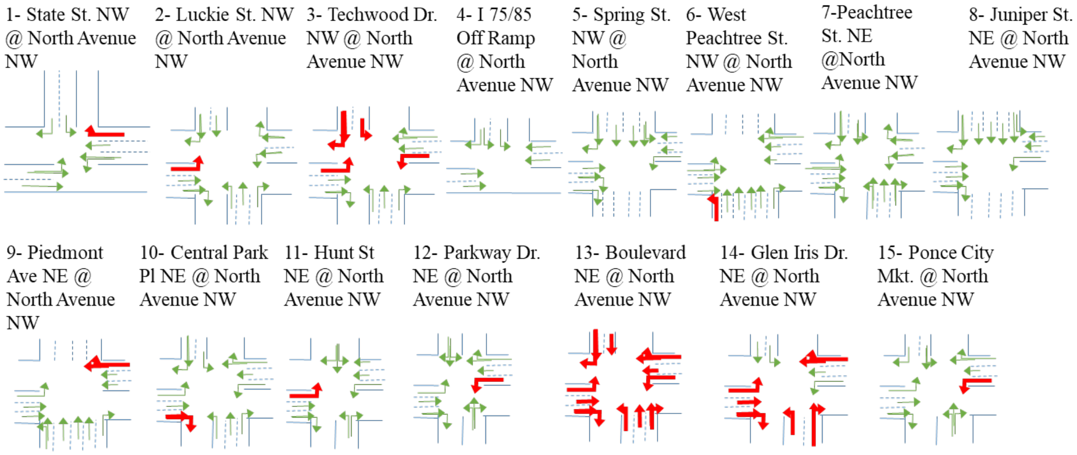


Figure 26: Missing volume pattern for 147 detectors over 24 hours on February 15, 2019.

Inspection of missing value patterns throughout the study period revealed that an intermittent data loss event nearly always includes all lanes of an approach, i.e., data are rarely received from some lanes on an approach while not on others. However, where data loss was permanent (i.e. no data were collected over the entire 112 days), data loss may incorporate all lanes on an approach or be isolated to a single approach lane, likely indicating either equipment failure or that the given lane was not detectorized. Figure 27 summarizes availability of detectors based on studying the volume data gap patterns.


1- State St. NW @ North Avenue NW
2- Luckie St. NW @ North Avenue NW
3- Techwood Dr. NW @ North Avenue NW
4- I 75/85 Off Ramp @ North Avenue NW
5- Spring St. NW @ North Avenue NW
6- West Peachtree St. NW @ North Avenue NW
7-Peachtree St. NE @North Avenue NW
8- Juniper St. NE @ North Avenue NW
9- Piedmont Ave NE @ North Avenue NW
10- Central Park Pl NE @ North Avenue NW
11- Hunt St NE @ North Avenue NW
12- Parkway Dr. NE @ North Avenue NW
13- Boulevard NE @ North Avenue NW
14- Glen Iris Dr. NE @ North Avenue NW
15- Ponce City Mkt. @ North Avenue NW

Note: The numbers present in the intersection names are associated with the intersection numbering provided in the Google Maps image shown in Figure 11.

Figure 27: Volume detection availability at the intersections, per lane, per movement. Permanently missing movements are identified in bold and red.

4.4.2. Signal Data Streams

Similar to the volume data-stream, a review of the signal data stream revealed message losses as well. Since the messages are used to update the status in the simulation whenever an indication change occurs, the lost messages can led to incorrect signal status indications in the simulation. Since the outages could be on the order of a few seconds to several hours, dropped messages could result in unrealistic timing pattern transitions as a missed change message could result in Vissim transitioning between non-sequential signal states, e.g., transitioning directly from the GREEN of one phase to the GREEN of the next, entirely skipping indication(s), etc. It is critical to recognize that the signals in the field did not likely experience these issues, instead the data loss is occurring between the field and the Vissim model.

The primary issues related to signal data loss can be listed as follows:

1. Message latency beyond negligible levels: usually the signal state messages are received within milliseconds of the actual occurrence in the field. However, instances were seen where the message delay was on the order of a signal cycle. As such, the delayed message causes a difference between the traffic signal state in the field and the status reflected in the simulation.
2. Intermittent message drops: due to intermittent communication failures in the wireless (cellular) network, there could be message drops. Dropped messages lead to a failure to update the status in the simulation and lead to an incorrect extension of the signal state in the simulation.
3. Extended message transmission failure: termination of message receiving process due to malfunctioning equipment (hardware) or crashing of communication routines may lead to an extended outage that can last a few hours to a few days, depending on the nature of the failure and the maintenance response.

4. System Architecture: while the model architecture has undergone robust testing the potential remains for errors or issues in the code, where unforeseen exceptions are not captured.

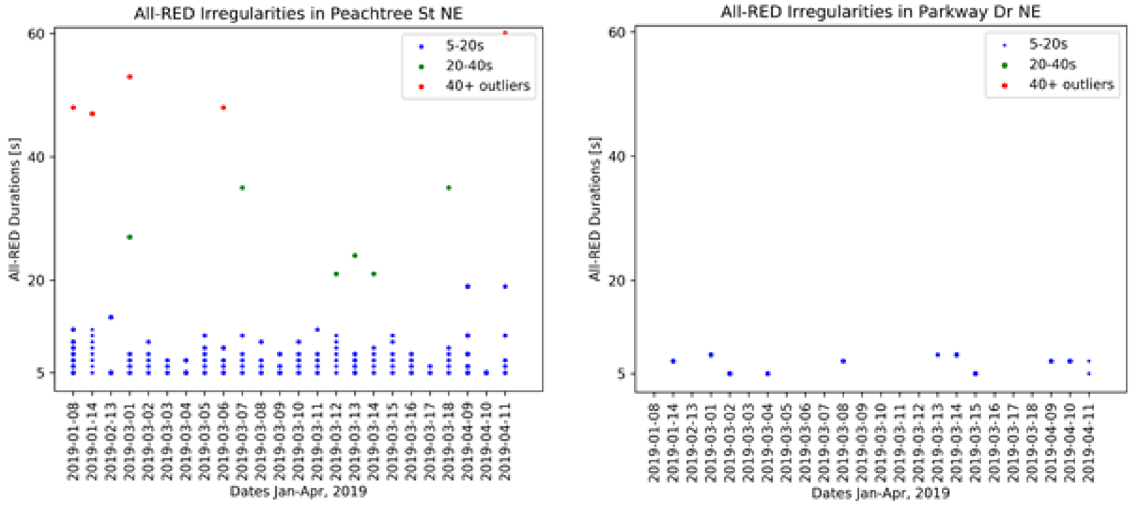
Loss of signal stream messages was seen to result in several potential errors in implementing the signal control in Vissim, including:

1. Long RED-Clearance, GREEN, AMBER, or RED
2. Missing GREEN, AMBER, or RED in a cycle
3. Missing block of signal information for a longer time period, from several cycles to hours.

The trends of signal data losses were studied over 49 different days from January, 2019 to May, 2019, with a mix of week days and weekends. The data was tested for integrity using some basic rules of acceptable durations of RED, AMBER, and GREEN.

4.4.2.1. All-RED Duration

Given the traffic signals in the network, All-RED durations of more than 5s were deemed likely incorrect and hence were tagged as anomalies. The 3D plots are helpful in visualizing the patterns of the anomalies and are available on the project website at: http://realtime.ce.gatech.edu/coa/sigplots/allred.html. It was found that I-75/85 Connector, Spring St., and West Peachtree St., have consistently high occurrences of long all-RED instances across all days. Also, after some maintnance updates in March, there has been an increased occurrence of anomalies in all-RED durations for the Spring St. intersection. When each intersection was inspected individually a wide variation in the occurance of all-RED irregularities (i.e. greater than 5s all-RED) was seen, as demonstrated by comparing Peachtree St. NW (Figure 28a) to Parkway Dr. NE (Figure 28). (See also http://realtime.ce.gatech.edu/coa/sigplots/allredselect.html)



(a) (b)

Figure 28: All-RED irregularities: (a) Peachtree St. NE and (b) Parkway Dr. NE

4.4.2.2. RED Duration

The 3D plots representing the anomalies related to RED durations are available on the project website at: http://realtime.ce.gatech.edu/coa/sigplots/red.html. In these plots, the RED durations below 400s are plotted at different times of the day for each intersection. Several instances showed longer RED duration than would be expected. In addition, RED durations of individual intersections at different times of the day can be inspected at: http://realtime.ce.gatech.edu/coa/sigplots/redselect.html. When inspected individually, several trends of irregularities were discovered related to RED durations. For example, one such irregularity trend was noticed for the eastbound movement of Peachtree St. NE (Figure 29), where since mid-April RED durations in excess of 400 seconds are seen at the end of the day. Other irregularities appear random and do not present any particular trends for this intersection. Similar issues are seen at other intersections.

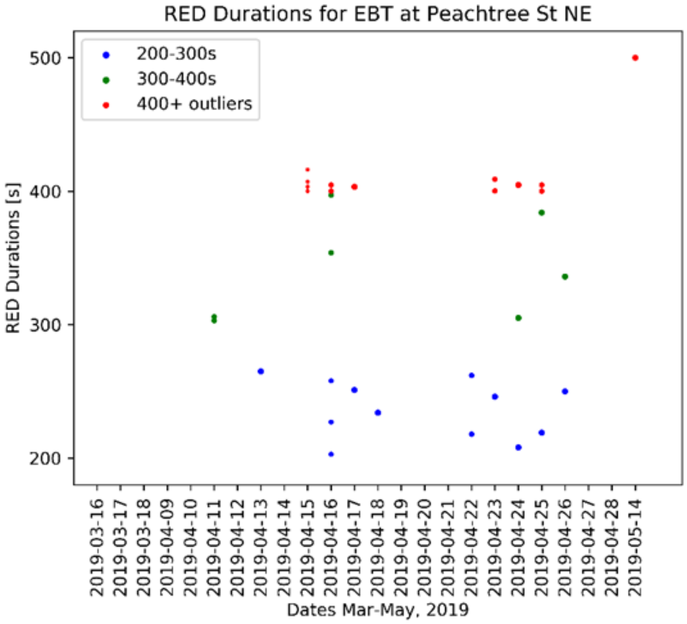
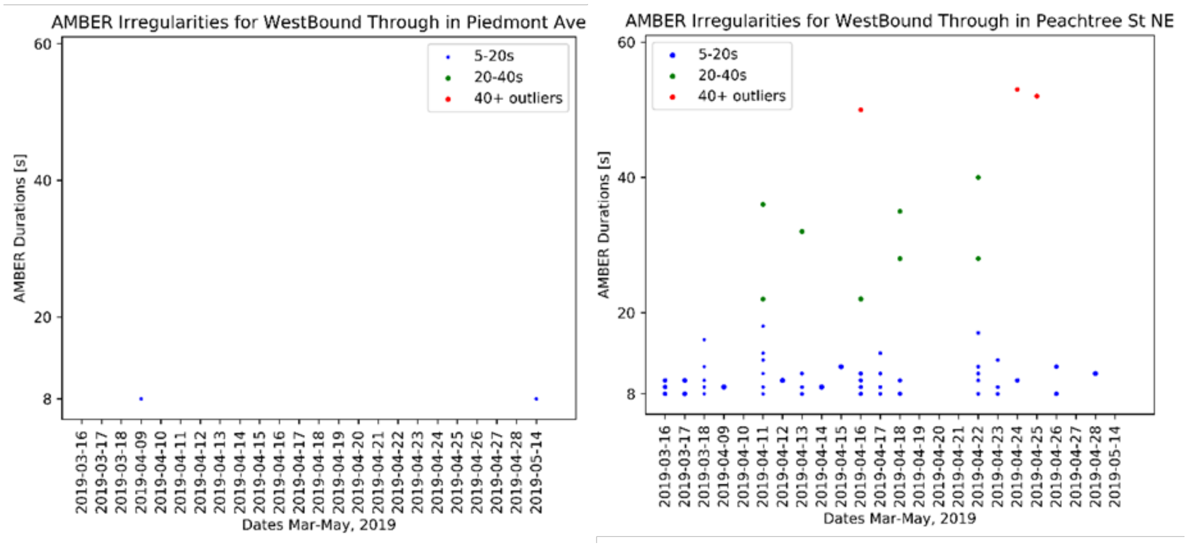


Figure 29: RED duration irregularities

4.4.2.3. AMBER Duration

The 3D plots of the anomalies for AMBER durations are available on the project website at: http://realtime.ce.gatech.edu/coa/sigplots/amberselect.html. It is observed that for AMBER anomalies, different intersections show different trends.

Plots were made for all intersections, for the instances in which the AMBER duration for any movement was more than 8 seconds. It was observed that some intersections had relatively fewer issues compared to others. For instance, when considering the AMBERs for the westbound through movements a few irregularities are seen on Piedmont Ave., whereas Peachtree St. NE has a significantly higher number of irregularities, mostly in late March and early April. Figure 30 provides sample data from the Piedmont Ave. and Peachtree St. NE intersections.



(a) (b)

Figure 30: AMBER duration irregularities: (a) Piedmont Ave., (b) Peachtree St. NE

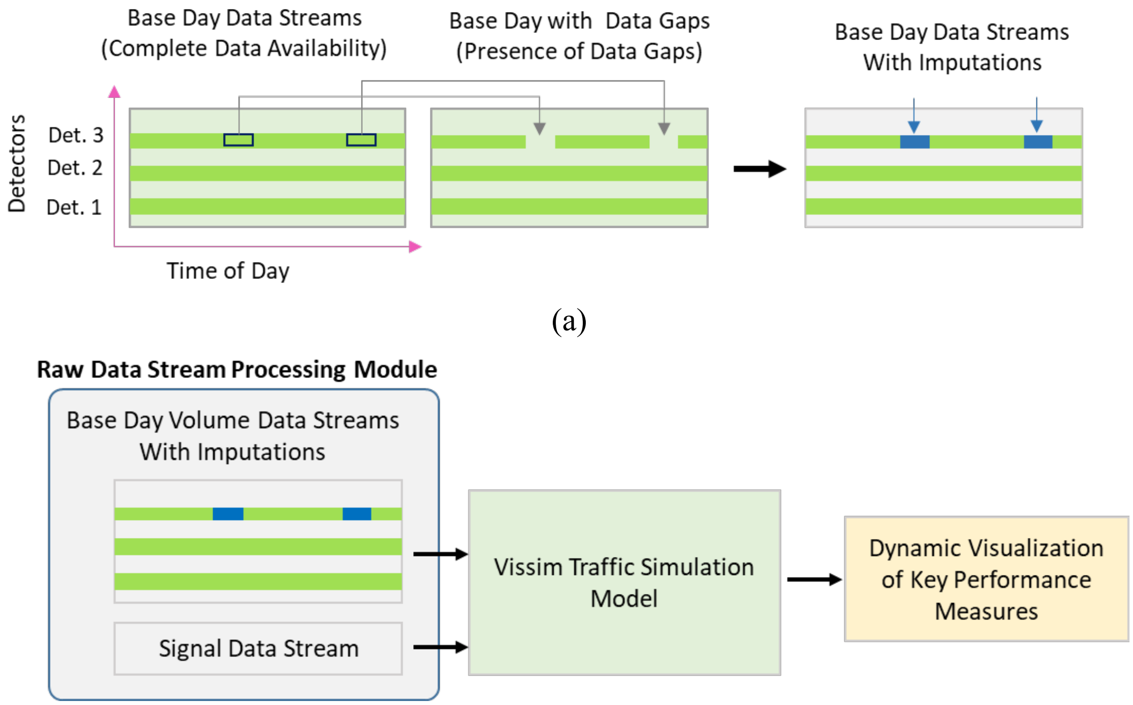
4.5. Sensitivity Analysis Experiment Methodology

Gaps in the data stream feeds to the simulation can result in underestimates in traffic volumes or unrealistic congestion due to incorrect signal control. While imputations on data gaps can be used to emulate a more realistic traffic scenario, it is also crucial to understand the potential impact of imputation errors on the simulation results. While data imputations are needed in both volume and signal data streams, this project initially focuses on understanding the impact of volume data imputations on model-generated performance measures. For this effort, signal messages were generated to infill missing data based on historic timings; however, future efforts will explore impacts of potential signal errors.

4.5.1. Experiment Design

The current data imputation method assumes that imputed data are drawn from historic data, and as such, may differ from actual field conditions. The current field data have no day with complete data, that is, all 112 days in the data set had both volume and signal timing outages, so a composite typical day with complete data was generated. Monday, March 18, 2019, was chosen as the *base day*, with missing signal and volume gaps imputed based on historic data, existing signal timing plans, alternate data sources (e.g., count data from a corridor development report), and any other available data sources. This *base day* is then considered to be "accurate" field conditions for the sensitivity analysis. In addition, 24-hour data loss patterns are generated based on the 112 days of field data. A random selection of five of the data loss patterns are applied to the *base day* to create five *base day with data gaps* scenarios. Data imputation is then undertaken for these scenarios. Three levels of volume imputation error are tested—20%, 50%, and 80% —with imputed volume errors inserted into the detection data streams for those detectors and times identified as having data loss in the given data loss pattern. The error levels are representative of the difference observed in volumes on the corridor in several locations over the 112 days. Ten replicate trials are completed for each *base day with data gaps* gap pattern at each error level, resulting in 160 total simulation trials (including 10 replicate trials of the *base day*). Key performance measures are then generated from the simulation tool for a *base day* and the *base day with data gaps*, allowing for an evaluation of the impact of volume data imputation errors. As the data are prepared *a priori* for this experiment the simulation was run faster than real-time to allow reduced processing time.

Figure 31 schematically shows the design of volume data stream imputations and the logic position in the real-time simulation model architecture. In the real-time simulation model architecture, data imputations are applied at the data input level prior to injecting the data into the simulation model. Thus, the imputed data will be formatted as the field data and the imputation method will have no dependency on the simulation model (Vissim in this experiment).



(a)

(b)

Figure 31: (a) *Base day* raw volume data stream to *base day* with imputed data, (b) Volume imputations for *raw data Streams processing module.*

4.5.1.1. *Modelling Likelihood of Missing Data on a Simulated Day Using Unsupervised Learning Method*

One of the key items in the experiment is the generation of data gap patterns. Data gap patterns are based on the 24-hour volume data streams received over 112 days spread across February, March, April, and June 2019, at 147 detector locations in the study corridor. As discussed, volume data for each detector are received in 6-minute aggregate vehicle counts. Over the 112 days, 29 detectors failed to provide any detection data. The remaining 118 detectors experienced at least one intermittent data loss, ranging from 6-minutes (a single bin) to an entire day. To analyze the data loss patterns a binary representation of the volume data was generated, where 0 represents volume data presence with a 6-minute bin and 1 represents volume data absence. Each detector, per day, was considered a sample for the cluster analysis. Thus, for the detectors that received at least some data there are 13,216 detector samples (i.e., 118 detectors x 112 days). These samples are clustered into groups to allow for a determination of the likelihood of different failure patterns. K-means unsupervised learning is used to cluster the 13,216 samples into groups based on selected key characteristics of the data gap pattern.

Missing Volume Pattern Grouping using K-means Clustering Algorithm: Each detector sample contains 240 binary values (i.e., number of 6 minute intervals in a day) representing the presence or absence of detection per interval. To find clusters with different presence and absence patterns, features are extracted that describe the primary characteristics of potential patterns. This feature engineering reduces the dimension of each sample from 240 to 7.

The following seven features were selected based on multiple clustering trials:

- Feature 1 - Total count of intervals without data over 24 hours: captures variation in number of absences
- Feature 2 - Average separateness between intervals without data, i.e., average separateness between absences: captures variation in spread between absence occurrences
- Feature 3 - Number of intervals without data within a cluster (i.e., more than one consecutive interval without data): captures number of absence intervals within groups
- Feature 4 - Maximum consecutive string of intervals without data in a 24-hour period: captures variation in maximum cluster of absences group size
- Feature 5 - Median monsecutive string of intervals without data in a 24-hour period: captures variation in median cluster of absences group size
- Feature 6 - Sum of intervals without data included in a string with 24-hour period: captures variation in total number of intervals with a group
- Feature 7 - Sum of positions of first interval with missing data point relative to start of day and last missing data point from end of day: captures variation in range in which absence occurrences are distributed within the 24 hr. period

The 24-hour data samples with 0 absences (no data loss) and 240 absences (complete data loss) are separated from the data set based on the feature total count of intervals without data over 24 hours. K-means clustering is applied on the remaining instances. To reduce dimensions and to reduce correlation between variables, principal component analysis (PCA) is performed on the clustering dataset. The dataset contains 13,216 instances, where each instance has seven features. The dataset is standardized over each of the seven features before conducting PCA by applying equation (1) on each data point of the seven features. Thus, each feature in the standardized data set has mean 0 and variance 1.

$$x_{standardized} = \frac{x_{original} - \mu_{feature}}{\sigma_{feature}} \quad (1)$$

PCA can provide a low dimension orthogonal subspace in which the variance of the projected data is maximized (Hotelling 1933). To find this subspace, PCA involves eigen-decomposition of the covariance matrix of the data. The eigenvalues provide variance of the projected data on the principal component along the corresponding eigenvector. Thus, the projected data has highest variance along the eigenvectors that corresponds to the highest eigenvalues. Results from PCA conducted on the dataset are summarized (Bishop 2006).

Eigenvalues from PCA:

$$[\lambda_1 \quad \lambda_2 \quad \lambda_3 \quad \lambda_4 \quad \lambda_5 \quad \lambda_6 \quad \lambda_7\,] = [\,4.39 \quad 1.66 \quad 0.67 \quad 0.21 \quad 0.04 \quad 0.01 \quad 0.001\,]$$

Percent of variance explained by the seven eigenvectors:

$$[0.6271 \quad 0.2372 \quad 0.0960 \quad 0.0304 \quad 0.0061 \quad 0.0026 \quad 0.0002]$$

The percent of variance explained by the first three eigenvalues are approximately 63%, 24%, and 10%. PCA results show that ~ 97% of variance in data is captured by the first three eigenvectors space, that is, the first three principal components (PCs). Table 2, the eigenvector matrix provides the vector representation of the three principal components along with weights of each feature on their respective PCs.

Table 2: Eigenvector matrix vector representation of the first three principal components

$$\left[\begin{array}{ccccccc} \boldsymbol{W1} & \boldsymbol{W2} & \boldsymbol{W3} & \boldsymbol{W4} & \boldsymbol{W5} & \boldsymbol{W6} & \boldsymbol{W7} \\ 0.46918118 & -0.3278184 & 0.24909679 & 0.44031693 & 0.4150821 & 0.45967952 & -0.18132154 \\ 0.04402246 & 0.47814207 & -0.43822556 & 0.28903463 & 0.33950239 & 0.18195218 & 0.58778487 \\ 0.05966025 & 0.13829284 & 0.76745451 & -0.08249622 & -0.19575754 & 0.10498177 & 0.57635189 \end{array}\right]$$

The clustering dataset is transformed into the three PCs space, thus, each data instance has only three features, its value on PC1, PC2, and PC3. Each PC is a weighted linear combination of features. Thus, projecting data points in the feature space of the three PCs the contribution of variation in data point by all the features will be captured. It can be observed that some features that are highly weighted on PC-1 have a lower weight value contribution than the other two PCs, such as features 1, 4, and 6. The clusters primarily reflect the number of intervals without data, indicating the first PC is reflecting a general likelihood of missing data. Feature 1, 4, 5, and 6 contribute highly to PC-1, which explains the 63% variance in data. PC-2, with significant weighting on features 2, 3, and 7, appears to reflect the spread of the intervals missing data. And finally, PC-3 appears to reflect the size of the cluster of intervals missing data. By transforming standardized data points into feature space of the three principal components, the contribution to variation in data points by all features is captured to a total of 97%.

The K-means clustering algorithm is applied to the transformed dataset. In this algorithm, the value of K, the number of clusters is set *a priori*. First, K random data points are chosen as centers of the K clusters. Next, each remaining data point is allotted to the cluster with the nearest chosen cluster center. After all data points are allotted to a cluster, new cluster centers are calculated. Based on the new cluster centers the data assignment process is then repeated, with all data points newly assigned to the nearest new cluster center. This process is repeated until the cluster centers stabilize and are not changing between repetitions. The objective function being minimized through this iterative algorithm is presented in equation (2) below. This value is the sum of within cluster sum of squared distances. That is, the objective function is the sum of squared Euclidean distances of data points from the cluster center to which they are assigned.

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2 \quad (2)$$

Where,

$n = 1, 2, 3, \dots, N$ is notation for N data points
x_n = n data point
$k = 1, 2, 3, \dots, K$ is notation for K clusters
μ_k = center (mean) of cluster K
$r_{nk} = \begin{cases} 1 & if\ x_n \in k \\ 0 & otherwise \end{cases}$

The two steps in the iterative process to minimize the objective function are: 1) to find a cluster assignment for each data point with fixed cluster centers that minimizes J, and 2) to find cluster center that minimizes J keeping the assignment of data point to clusters fixed. The K-means algorithm converges to local minimum for the objective function (Bishop 2006).

The number of clusters (K) is initiated based on results from the elbow method. The sum of squared distances of data points to their closest cluster center (inertia) is evaluated for varying values of the number of clusters (K). The point of inflection suggests a value for K. The highest K value after which the sum of the squared distance values is not significantly reduced is chosen, Figure 32 shows the graph obtained from elbow method. The K value after which sum of squared distance values does not drop significantly is observed in the graph to be 6 or 7. After looking at cluster results, 7 is chosen.

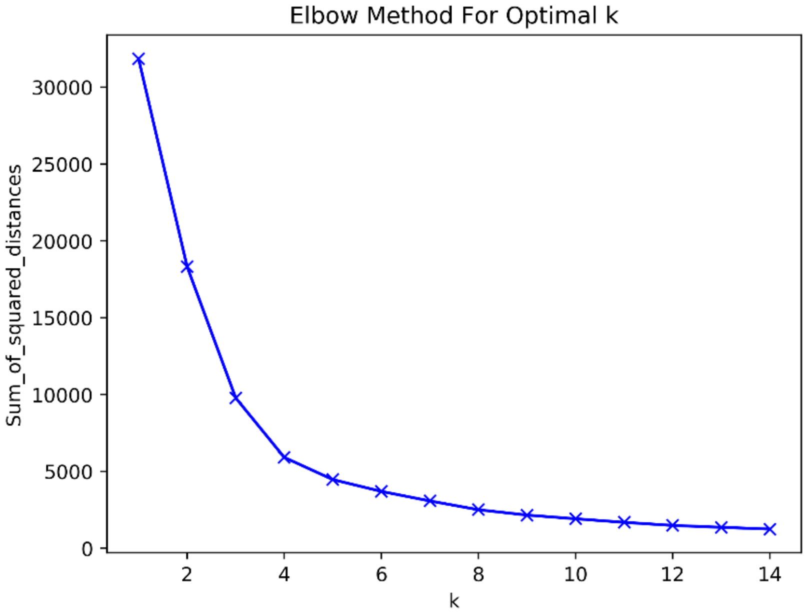


Figure 32: Elbow method to suggest K value. Sum of squared distances versus number of clusters.

Seven clusters are initialized and formed based on the K-means clustering algorithm. Similarity in two of the seven clusters is noticed and hence, they are merged into one single cluster. Thus, a total of eight clusters are created, where one cluster includes detector samples with no data loss, another cluster includes detector samples with a permanent data loss pattern, and the remaining six clusters are formed using clustering analysis. Figure 33 shows two of the clusters.

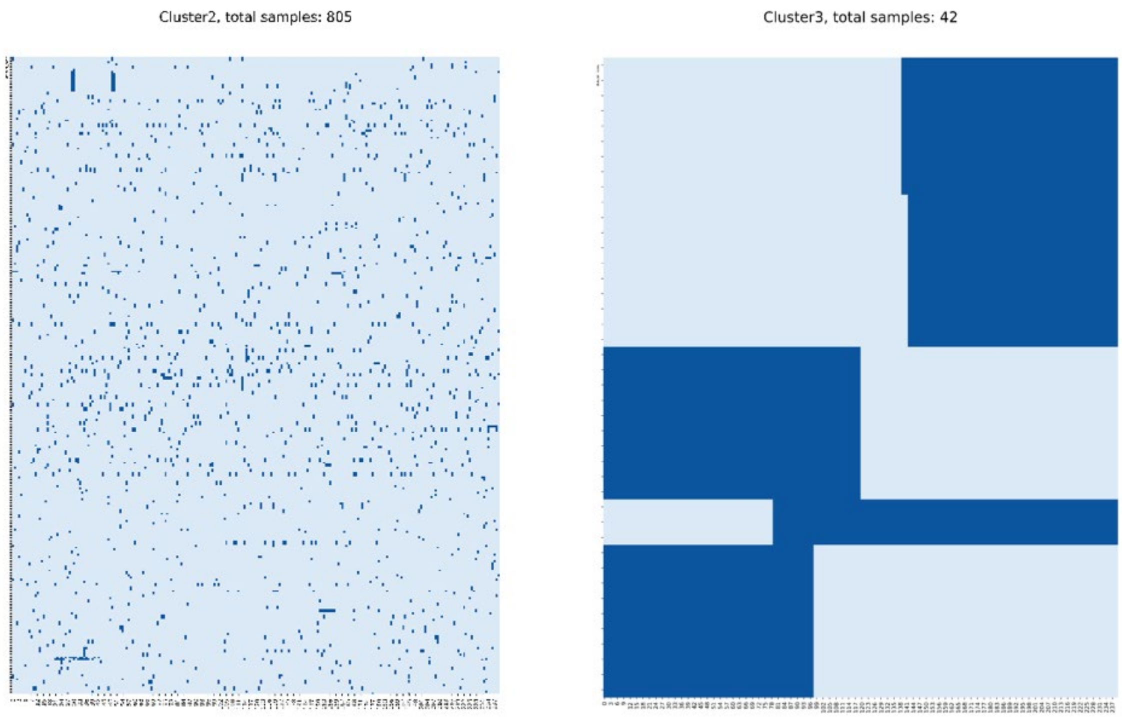


Figure 33: Two clusters of 24-hour data gap patterns with dark blue blocks as data loss intervals.

Figure 34 shows the K-means cluster assignment on the data points projected in the 3-PC space.

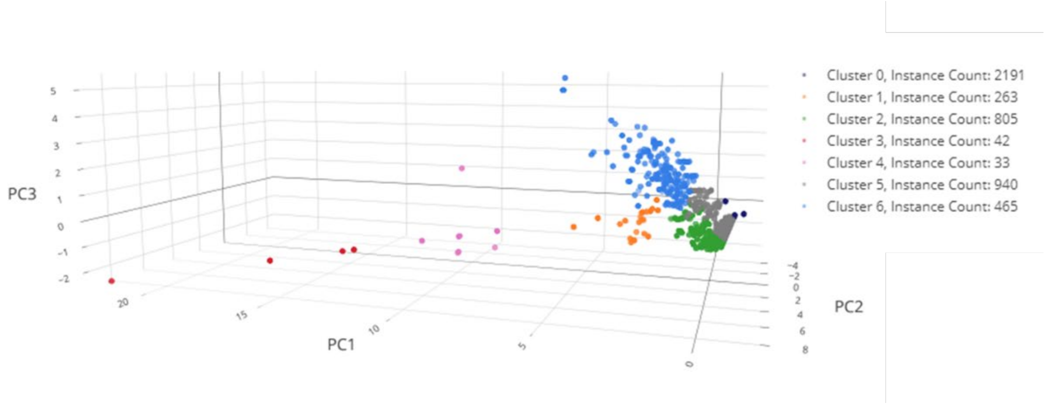


Figure 34: Clusters of data points projected in the 3-PC space.

<u>Missing-volume Pattern Generation</u>: The missing-volume pattern for a 24-hour day is generated by assigning each of the 147 detectors to a cluster, sampling a data loss pattern from the assigned cluster, and assigning that pattern to the detector. Detectors are randomly assigned to clusters with a likelihood based on the percentage of the 13,216 samples within each cluster. In addition, when assigning detectors to an intermittent data loss pattern, all detectors on an approach are considered together, as observations from the field data showed that in nearly all cases all detectors on an approach would exhibit the same intermittent data loss pattern. It was further noted that permanent data loss could be

observed at a single detector or at the intersection approach level. To implement assignment of detectors to a data loss pattern (Figure 35), first, a set of approaches with complete data loss are randomly assigned from all available approaches in the corridor, based on the likelihood of an approach having complete data loss, observed to be 0.13 in the 112-day sample. Then, from the remaining approaches, the approaches with complete data loss at a turn lane are randomly assigned, again based on field observation of 0.23 in the 112-day sample. Finally, all remaining detectors are randomly assigned a cluster and a missing-volume pattern.

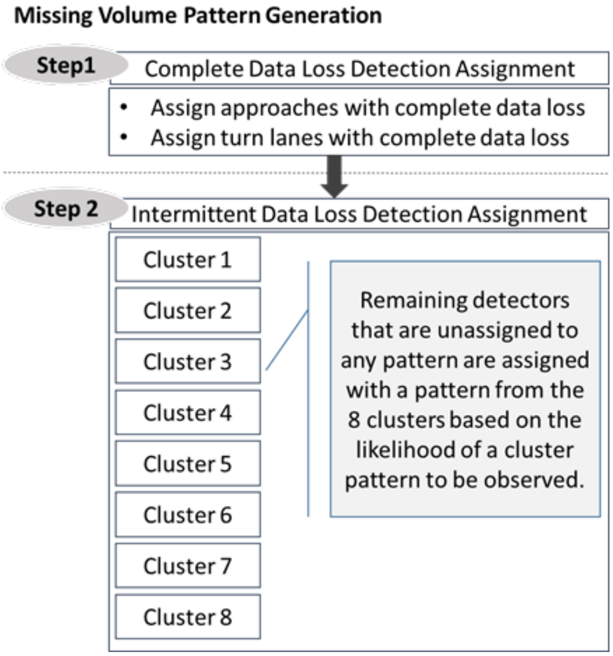

Missing Volume Pattern Generation
Step1
Complete Data Loss Detection Assignment
• Assign approaches with complete data loss
• Assign turn lanes with complete data loss
Step 2
Intermittent Data Loss Detection Assignment
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Cluster 8
Remaining detectors that are unassigned to any pattern are assigned with a pattern from the 8 clusters based on the likelihood of a cluster pattern to be observed.

Figure 35: Missing volume data pattern generation methodology.

4.5.1.2. *Simulation Experiment Implementation*

The simulation experiment involves three primary steps: 1) generate missing volume pattern, 2) generate *base day with data gaps* using the missing-volume pattern generated in the previous step, and 3) generate *base day with data gaps* with imputed data.

Simulation runs are conducted for a 3-hour PM peak period (3 PM – 6 PM). For five different missing volume patterns, the average travel time of a vehicle on a route for three values of error in data imputation – 20%, 50%, and 80% – is compared with that of 0% error in data imputation, implying complete data availability. Thus, for each of five missing value patterns, runs are conducted for 0%, 20%, 50%, and 80% value of error in data imputation for 10 random seeds.

4.5.1.3. Data Collection Routes and Detector Outages

The effect of 20%, 50%, and 80% error in data imputation (*base day with data gaps*) versus the *base day* on travel time is studied by conducting 10 different replicate trials for 5 different data loss patterns. Six routes along the mainline of the study corridor are selected for travel time comparison. These are:

1. Route 57: Eastbound State St. NW to Ponce City Mkt. (full corridor length)
2. Route 58: Westbound Ponce City Mkt. to State St NW (full corridor length)
3. Route 59: Eastbound State St. NW to Spring St. NW
4. Route 60: Eastbound West Peachtree St. to Hunt St. NW
5. Route 61: Westbound Hunt St. NW to West Peachtree St. NW
6. Route 62: Westbound Spring St. NW to State St. NW

Along with these selected six routes, the effect of data loss on network entry approaches is also studied. For example, missing-volume pattern 1, shown in Figure 36, has three boundary intersection approaches and six internal intersection approaches with permanent data loss on all lanes, and six boundary approaches and seven internal approaches with intermittent data loss on all lanes. In these experiments, potential error in volume imputation values for the data loss on boundary approach lanes is inserted by adjusting the base volume by 20%, 50%, or 80%. As the simulation is providing vehicle movement once a vehicle enters a boundary link, similar errors may not be applied directly to internal links with detection outages. However, to reflect the potential impact of volume imputation values on internal links the turn movement percentages for the missing lane configurations are increased by 5%, 10%, and 15% for the data imputation error scenarios of 20%, 50%, and 80%, respectively. This reflects that while internal detector data outages do not directly impact the absolute volume through an intersection, they may impact the assigned "volume splits." For example, if the left-only turn lane is missing at an internal approach, the left turn percentage at this approach is increased by 5%, 10%, or 15% for the respective error in volume data imputation scenario of 20%, 50%, and 80%.

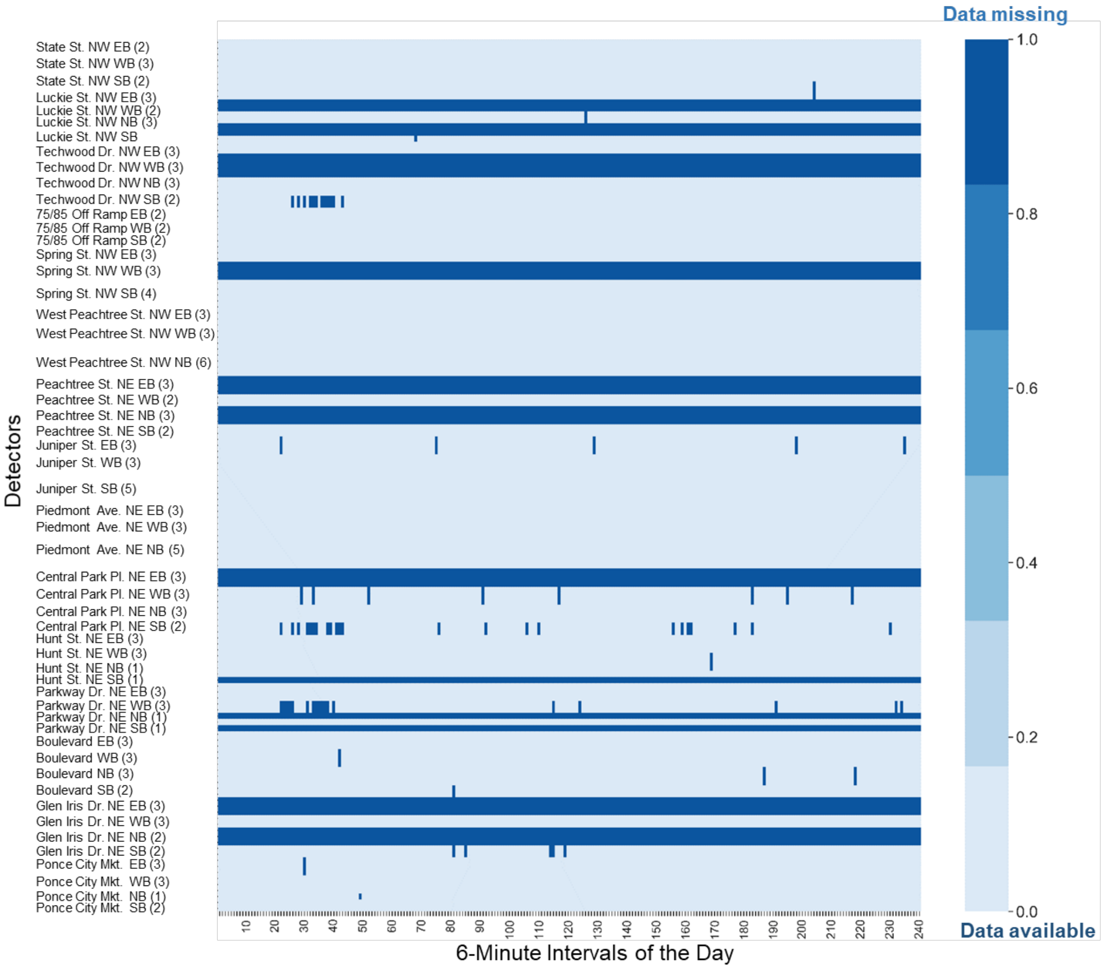


Figure 36: Missing volume Pattern 1 generated for the sensitivity experiment.

Table 3 lists the boundary and internal approaches with permanent or intermittent data loss for each of the five patterns.

Table 3: Summary of Boundary and Internal Approach Lanes with Permanent or Intermittent Data Loss for the Five Detector Outage Patterns.

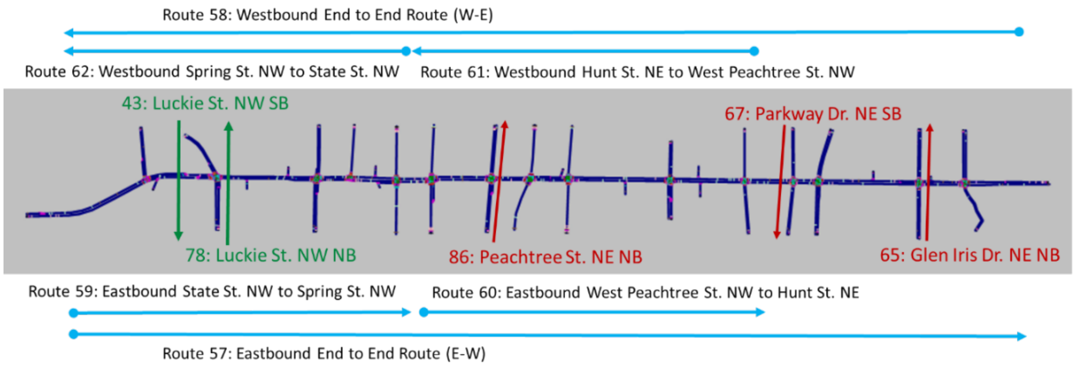
	Permanent Data Loss				**Intermittent Data Loss**			
	Boundary Approach/Lane		**Internal Approach/ Lane**		**Boundary Approach/Lane**		**Internal Approach/Lane**	
Pattern	**Boundary Approaches with Permanent Data Loss on All Lanes**	**Boundary Approaches with Permanent Data Loss on at least One Lane**	**Internal Approaches with Permanent Data Loss on All Lanes**	**Internal Lane Approaches with Permanent Data Loss on at least One Lane**	**Complete Entry Approaches with Intermittent Data Loss on All Lanes**	**Entry Lane Approaches with Intermittent Data Loss on at least One Lane**	**Complete Internal Approaches with Intermittent Data Loss on All Lanes**	**Internal Lane Approaches with Intermittent Data Loss on at least One Lane**
Pattern 1	PTST (NB), PWDR (SB), GIDR (NB), HNST (SB)	LKST (NB), LKST (SB),	LKST (WB), TWDR (WB), SPST (WB), PTST (EB), CPPL (EB), GIDR (EB)	TWDR (EB), PWDR (WB), GIDR (WB)	TWDR (SB), BLVD (NB), BLVD (SB), GIDR (SB), PCM (NB), CPPL (SB)	LKST (NB)	LKST (EB), JNST (EB), CPPL (WB), HNST (WB), PWDR (WB), BLVD (WB), PCM (EB)	PWDR (WB)
Pattern 2	CPPL (SB), HNST (NB), HNST (SB), PWDR (NB), PWDR (SB), GIDR (NB)	PTST (SB), PCM (SB)	SPST (WB), PDAV (WB), CPPL (EB), BLVD (EB), BLVD (WB)	LKST (EB), JNST (EB), PWDR (WB)	LKST (NB), LKST (SB), PDAV (NB), BLVD (SB)	PTST (SB), PCM (SB)	STST (EB), STST (WB), LKST (WB), TWDR (WB), OFRP (SB), SPST (EB), GIDR (WB), PCM (EB)	JNST (EB)

Pattern 3	SPST (SB), CPPL (NB), GIDR (SB), PCM (SB)	STST (SB), TWDR (NB), OFRP (SB), PTST (NB), JNST (SB), CPPL (SB)	PTST (EB)	STST (EB), LKST (EB), TWDR (EB), CPPL (EB), PWDR (EB), GIDR (WB)	TWDR (SB), PTST (SB), PDAV (NB), HNST (NB), HNST (SB), BLVD (SB), PCM (NB)	STST (SB), TWDR (NB), JNST (SB)	SPST (WB), PDAV (WB), PWDR (EB), BLVD (EB), GIDR (EB)	CPPL (EB), GIDR (WB)
Pattern 4	SPST (SB), PWDR (NB), GIDR (SB)	TWDR (SB), WPTST (NB), PTST (NB), PCM (SB)	STST (EB), PDAV (WB), CPPL (EB), CPPL (WB),	SPST (EB), JNST (WB), HNST (WB), PCM (EB), PCM (WB)	STST (SB), TWDR (NB), PTST (SB), CPPL (SB), BLVD (NB), BLVD (SB), GIDR (NB),	TWDR (SB), PTST (NB)	OFRP (WB), PTST (WB), PWDR (WB), BLVD (WB), GIDR (WB)	JNST (WB), HNST (WB), PCM (WB)
Pattern 5	LKST (NB), LKST (SB), BLVD (SB), PCM (NB), HNST (NB), HNST (SB)	OFRP (SB), PTST (SB), JNST (SB), SPST (SB)	SPST (EB), JNST (WB)	STST (EB), STST (WB), LKST (EB), TWDR (EB), TWDR (WB), WPTST (WB), HNST (WB)	STST (SB), PTST (NB), PDAV (NB), BLVD (NB), PCM (SB)	–	OFRP (EB), OFRP (WB), WPTST (EB), PTST (EB), PWDR (WB), BLVD (EB), GIDR (EB), GIDR (SB), PCM (EB)	TWDR (EB), WPTST (WB)

Key for Intersection Abbreviations Used in this Table	1. State St. NW @ North Avenue NW	STST
	2. Luckie St. NW @ North Avenue NW	LKST
	3. Techwood Dr. NW @ North Avenue NW	TWDR
	4. I 75/85 Off Ramp @ North Avenue NW	OFRP
	5. Spring St. NW @ North Avenue NW	SPST
	6. W. Peachtree St. NW @ North Avenue NW	WPTST
	7. Peachtree St. NE @ North Avenue NW	PTST
	8. Juniper St. NE @ North Avenue NW	JNST
	9. Piedmont Ave NE @ North Avenue NW	PDAV
	10. Central Park Pl. NE @ North Avenue NW	CPPL
	11. Hunt St. NE @ North Avenue NW	HNST
	12. Parkway Dr. NE @ North Avenue NW	PWDR
	13. Boulevard NE @ North Avenue NW	BLVD
	14. Glen Iris Dr. NE @ North Avenue NW	GIDR
	15. Ponce City Mkt. @ North Avenue NW	PCM

In Figure 37, the mainline routes studied for all patterns are shown on a network schematic. Figure 37also shows the five side-street routes studied for Pattern 1. For all patterns, side-street through-vehicle routes associated with the boundary approaches that observe permanent volume data loss for one or all lanes are studied. (For Pattern 1, Hunt St. NE (SB)), which experiences detector outage, has no through vehicles; hence, a side street route is not included.) The side-street route numbers associated with Pattern 1 are:

1. Route 43: Luckie St. NW SB (left turn only)
2. Route 78: Luckie St. NW NB (shared thru-right)
3. Route 86: Peachtree St. NE NB (all approach lanes)
4. Route 67: Parkway Dr. NE SB (all approach lanes)
5. Route 65: Glen Iris Dr. NE NB (all approach lanes)


Route 58: Westbound End to End Route (W-E)
Route 62: Westbound Spring St. NW to State St. NW
Route 61: Westbound Hunt St. NE to West Peachtree St. NW
43: Luckie St. NW SB
67: Parkway Dr. NE SB
78: Luckie St. NW NB
86: Peachtree St. NE NB
65: Glen Iris Dr. NE NB
Route 59: Eastbound State St. NW to Spring St. NW
Route 60: Eastbound West Peachtree St. NW to Hunt St. NE
Route 57: Eastbound End to End Route (E-W)

Note: Side street approaches that observed permanent loss on all lanes are besides red arrow and the side street approaches that observed permanent data loss at one lane are shown besides green arrow

Figure 37: Routes on which vehicle travel times are studied for Pattern 1 detector outages.

4.6. Results

Variation in mean and 85th percentile vehicle travel times for the ten replicate simulation runs for the *base case* (0%), and the *base case with gaps* (20%, 50%, and 80%) is studied. Similar trends are observed in both cases. For missing-volume Pattern 1, the variation in 85th percentile vehicle travel times for the 10 replicate simulation runs for the *base case* (0%), and the *base case with gaps* (20%, 50%, and 80%) are presented in the Figure 38 boxplot. For each of the six mainline routes, the boxplot presents the variation in the mean travel time of all vehicles that complete the given route, for each of the 10 replicate runs. An upward trend in travel times is observed from 0% to the 20%, 50%, and 80% error on most of the studied routes. The effect of error in data imputation on travel time is greater on the eastbound end-to-end route than the westbound end-to-end route. Route 59 and 60 are subparts of the eastbound end-to-end Route 57. Both the western half (Route 59) and eastern half (Route 60) contribute to the overall increase from west-to-east (Route 57). An upward trend in the mean travel time is observed on the side street Routes 67, 78, and 86, while Routes 43 and 65 do not have an increasing travel time with increasing errors. The lack of impact on Route 43 (Luckie St. NW (SB)) is likely as result of base traffic volumes being close to saturated condition; thus, additional travel time was experienced outside the travel time trap (vehicles were not able to enter the link), while Route 65 (Glen Iris Dr. NE (NB)) observed a very low hourly volume (approximately 110 vph) for the base case, resulting in a minimal travel time increase even given the increasing error percentage.

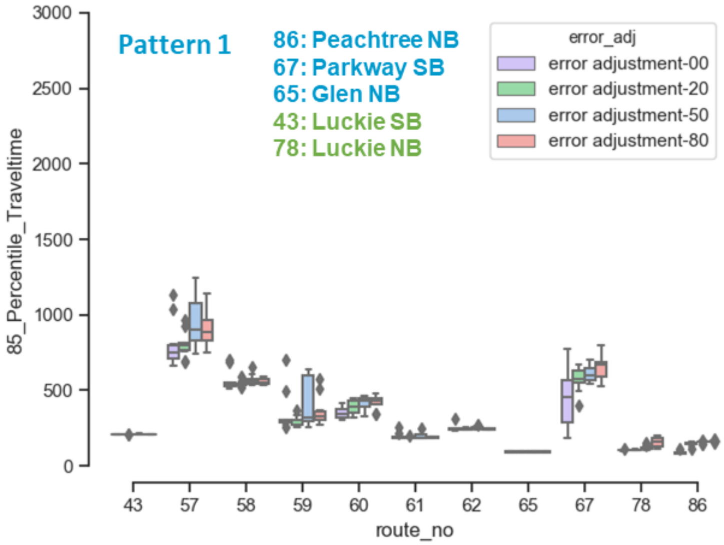


Figure 38: Boxplot of 85th percentile vehicle travel times for data imputation error cases, for Pattern 1 mainline and side street routes.

Figure 39 visualizes the mean simulation vehicle input count at the side streets for the error in data imputation cases – 20%, 50%, and 80% – in comparison to the *base case*. With the exception of Luckie St. NW (SB), which is at near-saturation state in the *base case*, the volume input counts at other approaches show an increase.

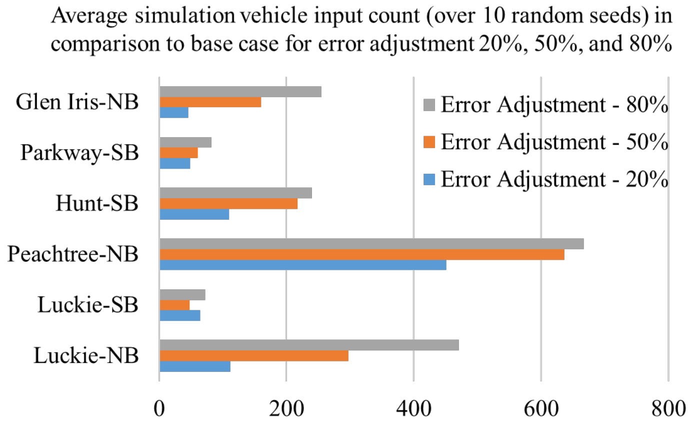
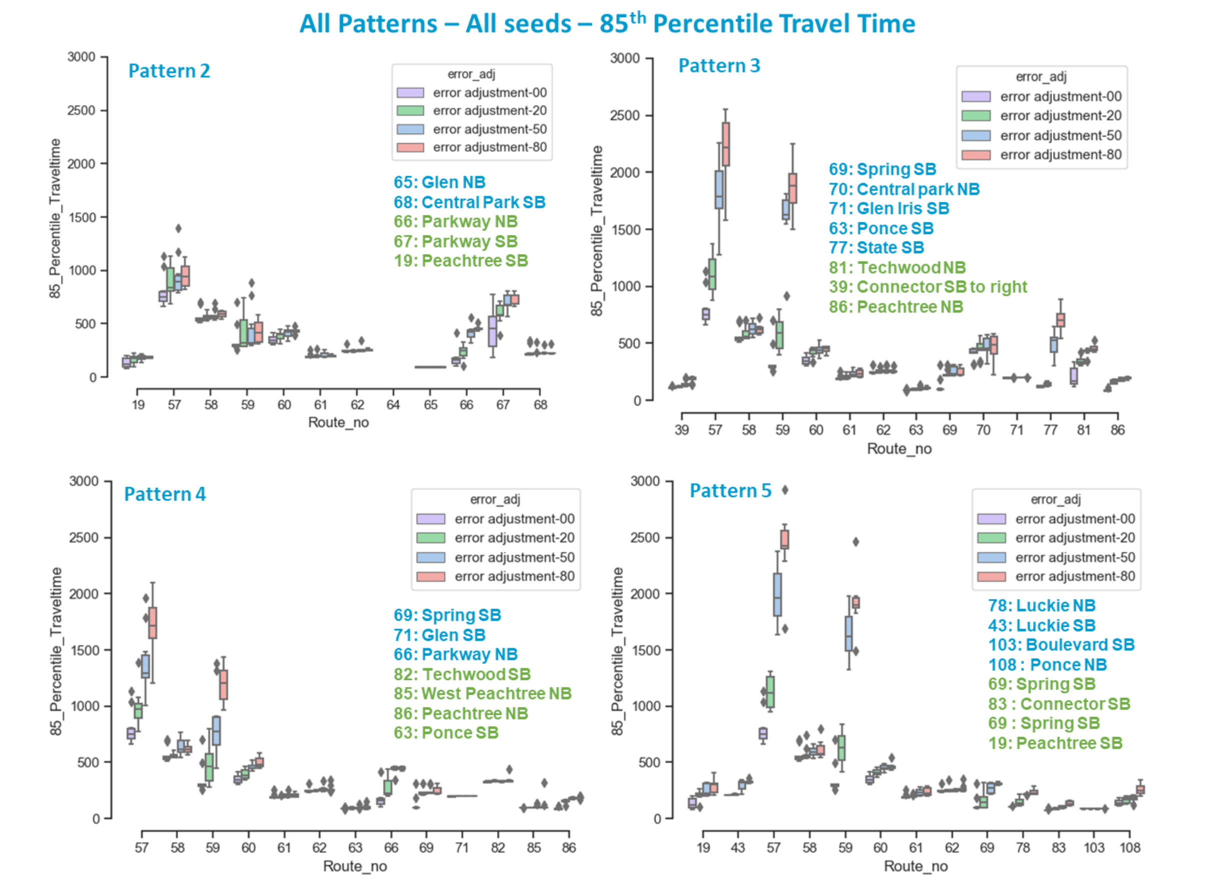


Figure 39: Mean simulation vehicle input count for data imputation error cases in comparison to the *base case* on the side-street routes that observed permanent data loss in Pattern 1.

Figure 40 shows the travel time impact of imputed volume data on approaches that observe permanent data loss, as well as the mainline routes for Patterns 2 through 5. For brevity, detailed results are not written for each, but findings from these patterns will be presented in the discussion section.



Common-to-all Patterns Simulation Network Travel Time Routes Studied

- 57: Eastbound end to end
- 58: Westbound end to end
- 59: Eastbound State St. to Spring St.
- 60: Eastbound W. Peachtree St. to Hunt St.
- 61: Westbound Hunt St. to West Peachtree St.
- 62: Westbound Spring St. to State St.

Figure 40: Variation in 85th percentile travel time values at the mainline and side street routes for the five missing volume patterns.

4.6.1. Intermittent Data Loss

The impact of intermittent data loss on travel time is studied using a statistical test for a selected case. Among the five missing-volume patterns simulated, for the PM peak hour period (3 PM – 6 PM), Pattern 3 contained three entry approaches with intermittent volume data loss - Techwood Dr. NW (SB), Piedmont Ave. NE (NB), and Boulevard (SB) - making this a good scenario to investigate the impact of intermittent data loss. The Techwood Dr. NW (SB) volume data loss pattern had more frequent distributed data gaps; hence, it was selected to study the impact of intermittent data gap imputations on side-street travel time.

4.6.1.1. *Case Study: Impact of Intermittent Volume Data Gap Imputations at Techwood Dr. NW*

To study the impact of intermittent volume data gap imputations on travel time, result from two simulation scenarios are compared: 1) Pattern 3 with intermittent and permanent data loss – referred to as *with intermittent loss scenario*; and 2) Pattern 3 with only permanent data loss – referred to as *without intermittent loss scenario*. Pattern 3 with only permanent data loss comprises an additional 10 simulation runs (10 replicates of the permanent detection loss only pattern) in addition to the 160 already discussed. The comparison of these two scenarios allows for an isolation of the impact of intermittent data loss. Results from 10 random seed simulation runs for the two scenarios are compared for the three imputations error cases: 20%, 50%, and 80%.

The simulation time of 3 hours (3 PM to 6 PM) is binned into 6-minute intervals. For each detector error scenario, for each replicated run, for each time bin, the 85th percentile travel time is calculated. This results in a series of 30 (i.e., the number of bins) 85th percentile travel times across the 3 hours, for each replicate trial. These 30 values are then averaged, resulting in a single 85th percentile value for each replicate. Finally, the difference between the replicate value for the *with intermittent loss scenario* and *without intermittent loss scenario* is calculated, where the paired replicates have the same seed. Equation (3) provides the described 85th percentile difference between paired replicated trials. The results for all scenarios are shown in and Table 4.

$$Sample\ set\ for\ error\ adjustment\ 20\%\ case\ = \frac{\sum_{\forall timebins}\left(85th\ percentile\ tt_{seed,int,20\%}^{timebin}\right)}{No.\ of\ timebins} - \frac{\sum_{\forall timebins}\left(85th\ percentile\ tt_{seed,woint,20\%}^{timebin}\right)}{No.\ of\ timebins}\ \forall\ seeds \quad (3)$$

Where,

tt = travel time
timebin = simulation time bin such as (0, 360), (360, 720), etc.
seed= simulation trail seed, such as seed 21, seed 22, seed 23, etc.
int = result from *with intermittent loss scenario* run
woin = result from *without intermittent loss scenario* run

A t-statistics hypothesis test (see Table 4) is conducted for each error adjustment in the volume data imputation case to test if imputations on intermittent data loss statistically increase the travel time. A one-sample t-test for the mean is conducted to test the alternate hypothesis that the mean 85th percentile travel time difference for 10 samples will be greater than 0, with the null hypothesis that the mean 85th percentile difference is 0. In this test, the sample set includes values of all 30 time bins.

Table 4: Average 85th percentile difference of all time bins for 10 random seed runs for the three error imputation scenarios.

Seed	Error adjustment of 20%	Error adjustment of 50%	Error adjustment of 80
Seed 1	24.22	20.01	7.05
Seed 2	27.50	61.82	24.78
Seed 3	17.08	44.46	3.90
Seed 4	-33.71	13.14	32.67
Seed 5	0.54	14.22	11.49
Seed 6	14.95	-3.75	27.34
Seed 7	-23.24	-0.53	10.49
Seed 8	-21.13	8.84	24.27
Seed 9	10.47	17.76	60.35
Seed 10	2.82	32.67	75.22
Mean	*1.95*	*20.86*	*27.75*
t-test P-value	*0.291*	*0.005*	*0.002*
Significant level:0.05	*P value > 0.05*	*P value < 0.05*	*P value < 0.05*
	Fails to reject null hypothesis	**Reject null hypothesis**	**Reject null hypothesis**

For an error adjustment of 20%, for a 0.05 significance level, the t-test fails to reject the null hypothesis that there is no effect on travel times due to imputations in intermittent data loss. While for the higher error adjustments, 50% and 80%, the t-test results reject the null hypothesis in favor of the alternate hypothesis that travel time for the *with intermittent loss scenario* is higher than that of the *without intermittent loss scenario.* The hypothesis test results indicate that higher error in data imputation values can impact the simulation-generated travel time results at approaches with intermittent data loss patterns when the sample difference values for all time bins are considered. The data plotted in Figure 41 provide a visual confirmation of this finding.

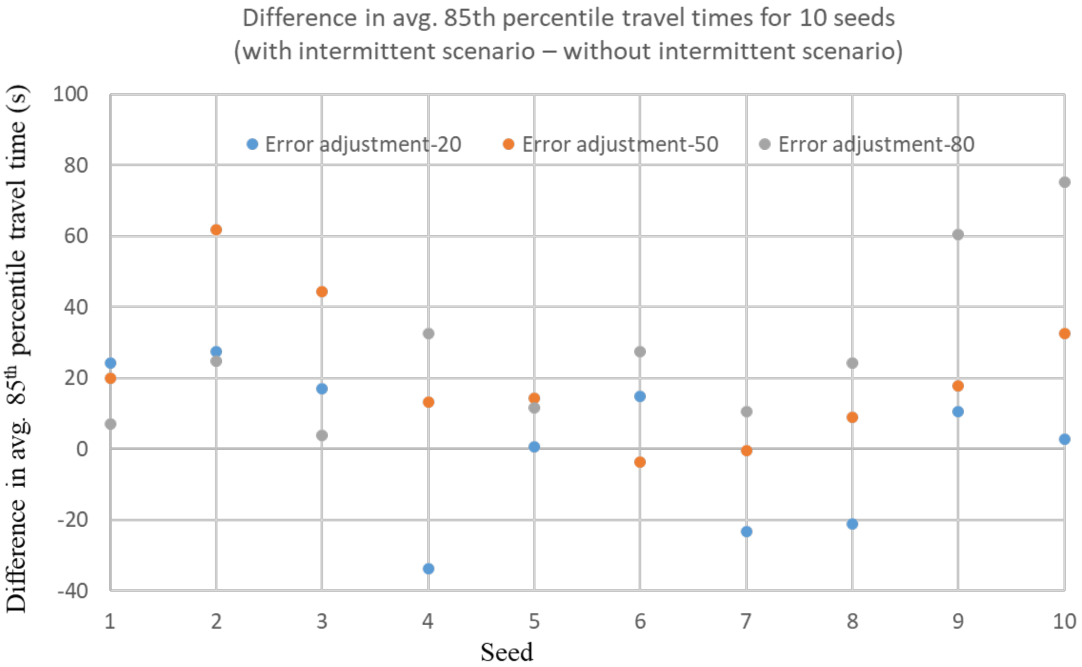


Figure 41: Variation in average 85 percentile travel time differences *with intermittent loss scenario* and *without intermittent loss scenario* of all time bins for different error in data imputation cases.

A similar hypothesis test is conducted to test if the travel times are significantly higher for time bins that follow the time bins with imputed values for the *with intermittent loss scenario* in comparison to that of the *without intermittent loss scenario.* That is, does the intermittent error potentially increase as the number of consecutive intervals with missing data increases? For this hypothesis test, the average 85th percentile difference in travel time is evaluated only for the six time bins that follow the time bins with intermittent data loss. Similar to the prior results, the hypothesis test t-test fails to reject the null hypothesis for the error adjustment 20% case while for the error adjustment 50% and 80% cases rejects the null hypothesis in favor of the alternate hypothesis that higher travel times are observed in time bins that follow after the imputed time bins. Unfortunately, the small sample size (six samples) likely limits the ability to statistically distinguish differences at lower error levels. Future analysis will expand the study period to include additional data samples.

4.7. Discussion

The results plotted in Figure 38 and Figure 40 for the five missing-volume patterns indicate higher sensitivity of travel time on the eastbound routes in comparison to that of the westbound routes, likely an indication of the eastbound direction of travel operating closer to capacity. This is seen in the simulation where at several intersections the simulation is unable to process the full traffic load created by the imputation errors. Mainline eastbound travel time increases are spread throughout the corridor, as seen by increases on both subroutes 59 (EB, west of Spring St.) and 60 (EB, east of Spring St.). However, one section of roadway may be more impacted; for instance, the western portion of North Ave. (Route 59) is the source of most of the eastbound increases in

Patterns 3, 4, and 5. Further, it is seen that the travel time increases on the eastbound end-to-end (Route 57) and the westbound end-to-end (Route 58) differ substantially across patterns, with Patterns 1 and 2 having less impact than Patterns 3, 4, and 5. The pattern travel time differences are due to the underlying data gaps and turn-movement errors specific to each pattern. For Patterns 1 and 2, the less-dramatic eastbound travel time increases indicate that approaches experiencing data gaps within these patterns have a lesser impact relative to the missing data in Patterns 3, 4, and 5. For example, the higher impact seen with Pattern 3 is hypothesized to be a result of data imputation on the Spring St. NW (SB) and 75/85 Off-Ramp/Connector approaches. While other approaches in the approach set for Pattern 3 (Central Park Pl. [NB], Glen Iris Dr. [SB], and Ponce City Mkt. [SB]) may contribute toward higher traffic in the eastbound direction, their contribution to the total higher traffic entering the simulation network is substantially less. In addition, high-volume approaches could be found to have minimal impact on overall performance where the approach had a base case demand volume near saturation. In these instances, the higher volumes in the error cases were unable to enter the network during the peak, minimizing the immediate impact of these errors. This does not imply that these errors do not influence traffic, simply that instead of resulting in more significant congestion during the current peak, they will have the effect of spreading out the peak, resulting in a longer period of congested traffic. It is not only total volumes that impact performance, but also position of the intersection. For instance, while Ponce City Mkt. (SB) does contribute traffic, its position as the east most intersection on the corridor lessens its impact on the eastbound traffic flow.

When considering side street performance, it is again observed that some side-street route travel times are more sensitive to data loss than others. Again, this is primarily a function of the base volume. For both very low and near-saturation volumes, the impact is minimal, while in the remaining cases the impact of the added volume can be significant. In interpreting this result, it is critical to recall that the volume errors occur due to failed receipt of detector data by the simulation platform. The field signal timing in all scenarios does not change, and it is for the correct traffic volume. Thus, the increased volumes in the simulation do not receive higher green times, as the signal timing is driven by the timing data from the field

4.8. Conclusion

The objective of this chapter was the creation of a dynamic data-driven simulation that leverages high-frequency connected data streams to derive meaningful insights about the current traffic state and near real-time corridor environmental measures. The developed simulation model emulates traffic on 2.3 miles of the North Avenue Smart Corridor in Atlanta, Georgia. For this chapter, the ability to stream the volume and signal data real-time was expanded to all modeled signalized intersections in the corridor. However, investigation of the data streams themselves revealed the presence of data gaps. Thus, the model architecture was enhanced to handle such data losses. This chapter provided an overview of the improved architecture and utilized sensitivity analysis to explore the impact of the volume data gap imputations on the key performance measures produced by the near-real-time data-driven simulation.

5. CONCLUSIONS

This report presents a near-real-time data-driven simulation of the North Avenue Smart corridor and investigates the sensitivity of simulated performance measures to imputations for data loss resulting from missing detector streams. The phase 1 hybrid model presented in Chapter 3, and the expanded model presented in Chapter 4, demonstrate the feasibility of the overall connected corridor simulation approach. Leveraging the high level of communications that are expected to be available on a Smart Corridor, extensive sensor deployment, and big data techniques, the real-time data-driven simulation architecture is developed to enable four primary tasks: 1) injection of real-time signal control and volume data that stream into the traffic simulation model through real-time raw data stream processing, 2) model execution through dynamic data-driven traffic simulation, 3) dynamic performance metric visualization, and finally 4) facilitation of efficient handling of transactions between modules using a data request transaction management module. Utilizing this architecture a demonstration of the calculation and dissemination of near-real time key performance measures is demonstrated.

Critically, this effort utilized real-time, field data streams. Thus, this effort was able to investigate issues related to the transfer of technology from the laboratory to the field. It was seen that a smart technology development cannot assume perfect conditions and must account for potential communication and data losses to ensure robustness. Even under the best circumstances communication and data challenges are to be expected. For instance, this study shows that the impact of data issues on the simulated performance measure error is, in part, dependent on the combination of intersection approaches experiencing data loss. This combination effect can be attributed to both the vehicle volumes observed at these approaches, as well as the ability of the approaches to process additional vehicles (i.e., approach capacity). The relative location of the intersection approaches with missing data in the corridor, as well as the travel path used in generating the measure, also influence performance measure accuracy. For instance, in Patterns 3 and 5, the close proximity of Spring St. (SB) and Connector (SB) contributed to the higher simulated performance measure errors. In addition, to the effects of the spatial outage patterns, the temporal outage patterns were also observed to have a measurable effect. The impact of intermittent data gap imputations on travel times was studied for the test case of Pattern 3, Techwood Dr. (SB). Significant differences in travel were seen for the 50% and 80% error cases. At higher data imputation errors (50% and 80%), travel time increases occur during the time bins with missing data, as well as those that immediately follow the imputed time bins.

Applying an understanding of the various factors influencing potential model errors from connected corridor missing detection data streams is crucial to assessing the impact of data imputation errors in the generated traffic performance measures. For instance, these insights could be used to develop confidence intervals on reported real-time model performance measures. This difference in impacts among intersections and approaches may also be used to prioritize intersection approaches for maintenance and monitoring, particularly where resources are limited. Thus, the identification and prioritization of data streams in a corridor can help provide a more robust system and a more efficient application and usage of smart technologies.

5.1. Recommendations and limitations

At their foundation "Smart" initiatives seek to convert data into actionable information. However, for the information to be reliable, the data generation must be resilient and the data processing needs to be robust. Thus, as cities desire to move into the smart technology paradigm it is critical that significant investment be made in efficient data management and handling techniques. In addition, cities will need to place high priority on the maintenance and equipment refresh cycle for those technologies that are incorporated into the Smart City platform to enable the high level of reliability demanded by real-time operations.

For smart, real-time applications utilizing these data streams, it is imperative to develop and understand the effectiveness of data imputation techniques, as even under the best of circumstances data streaming outages will occur. In addition, throughout this study it is assumed that the received data streams are a correct reflection of the field. However, these data streams likely contain sensor errors, and procedures need to be developed to evaluate and address these data shortcomings.

Specific to the architecture developed, additional improvements may also be considered. For instance, the current platform is limited to using a single simulation instance. The ability to incorporate multiple replicates should be considered. In addition, a larger network would likely require a distributed simulation approach to maintain faster than real-time performance. Ultimately, the goal of this research is to provide a predictive element, enabling the use of the model for real-time operational control decisions. Thus, future versions of the architecture will include rolling horizon, time warp, or a similar approach to near-term prediction and correction.

5.2. Next Steps

The next steps in this research would be to study and identify energy consumption and emissions values trends generated from the platform. The two key steps expected to be involved in this effort are to implement more advanced imputation methodologies for intermittent signal and volume data gaps and to design and conduct a simulation experiment to identify the trends in energy and emissions values. In addition, the traditional approaches of imputation using patterns derived from archival data from the same source, cross-technology imputation needs to be explored. Opportunities to leverage sparse connected vehicle data (that is available during the transition from non-connected to connected) to bridge the gaps in the signal volume data streams need to be investigated.

The transferability of the platform needs to be investigated. It is expected that the overall architecture is transferrable; however, specific calibrations and adjustments will likely be needed for each corridor. Finally, as stated, it is critical that the developed architecture be able to execute faster than real-time. As the number of intersections in the model increases this will becomes increasingly difficult to achieve within the current architecture. Ultimately, the current architecture needs to be converted to a parallelized architecture to support the scalability demands of larger models.

6. REFERENCES

(1) "SMARTEST" (2000). ITS, University of Leeds (GB).

(2) A Chronopoulos, A. T. and C. M. Johnston (1998). A real-time traffic simulation system.

(3) Agafonkin, V. (2019). "An open-source JavaScript library for mobile-friendly interactive maps." Retrieved December 20 2019, 2019, from https://leafletjs.com/.

(4) Allström, A., J. Barcelo, J. Ekström, E. Grumert, D. Gundlegård and C. Rydergren (2017). Traffic Management for Smart Cities.

(5) Amini, S., I. Gerostathopoulos and C. Prehofer (2017). Big data analytics architecture for real-time traffic control. 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS).

(6) Bishop, C. M. (2006). Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag.

(7) Brügmann, J., M. Schreckenberg and W. Luther (2013). Real-Time Traffic Information System Using Microscopic Traffic Simulation. 2013 8th EUROSIM Congress on Modelling and Simulation.

(8) Caltrans. "Connected Corridors." Retrieved 29 July, 2019, from https://dot.ca.gov/programs/traffic-operations/connected-corridors/overview.

(9) Chen, S. and L. Du (2017). "Simulation study of the impact of local real-time traffic information provision strategy in connected vehicle systems." International Journal of Transportation Science and Technology **6**(4): 229-239.

(10) Chowdhury, M., M. Rahman, A. Rayamajhi, S. M. Khan, M. Islam, Z. Khan and J. Martin (2018). "Lessons Learned from the Real-World Deployment of a Connected Vehicle Testbed." Transportation Research Record **2672**(22): 10-23.

(11) CityOfAtlanta. (2017). "NORTH AVENUE SMART CORRIDOR." Retrieved 29 July, 2019, from https://renewatlantabond.com/project/north-avenue-smart-corridor/.

(12) CityofAtlanta. (2018, May 22, 2018). "RENEWATLANTATSPLOST." Retrieved 24 May, 2018, from http://renewatlantabond.com/.

(13) Ding, Q., X. Wang, X. Zhang and Z. Sun (2010). "Forecasting traffic volume with space-time ARIMA model." Advanced Materials Research **156-157**: 979-983.

(14) Doecke, S., A. Grant and R. W. G. Anderson (2015). "The Real-World Safety Potential of Connected Vehicle Technology." Traffic Injury Prevention **16**(sup1): S31-S35.

(15) FHWA, U. (2019, 15 March 2019). "Simulating Connected Vehicle Technologies in Virtual Traffic Environments." Retrieved 29 July, 2019, from

https://highways.dot.gov/research-programs/exploratory-advanced-research/simulating-connected-vehicle-technologies-virtual.

(16) Ghosh, B., B. Basu and M. O'Mahony (2005). Time series modelling for forecasting vehicular traffic flow in Dublin.

(17) Glasmeier A, C. S. (2015). "Thinking about Smart Cities." Cambridge Journal of Regions, Economy and Society **8**(1).

(18) Goodall, N. (2013). Real-Time Prediction of Vehicle Locations in a Connected Vehicle Environment, Virgina Center for Transportation Innovation and Research**:** 55.

(19) Greco, H., Cresta A. (2015). A SMART PLANNING for SMART CITY: the concept of smart city as an opportunity to re-think the planning models of the contemporary city. International Conference on Computational Science and Its Applications.

(20) Guensler, R., H. Liu, Y. Xu, A. Akanser, D. Kim, M. P. Hunter and M. O. Rodgers (2017). "Energy Consumption and Emissions Modeling of Individual Vehicles." Transportation Research Record: Journal of the Transportation Research Board **2627**: 93-102.

(21) Guensler, R., H. Liu, Y. Xu, A. Akanser, D. Kim, M. P. Hunter and M. O. Rodgers (2017). "Energy Consumption and Emissions Modeling of Individual Vehicles." Transportation Research Record **2627**(1): 93-102.

(22) Henclewood, D., A. Guin, R. Guensler, M. Hunter and R. Fujimoto (2010). Real-time data driven arterial simulation for performance measures estimation. Proceedings of the 2010 Winter Simulation Conference.

(23) Hotelling, H. (1933). "Analysis of a complex of statistical variables into principal components." Journal of Educational Psychology **24**(6): 417-441.

(24) Lee, S. and D. B. Fambro (1999). "Application of Subset Autoregressive Integrated Moving Average Model for Short-Term Freeway Traffic Volume Forecasting." Transportation Research Record **1678**(1): 179-188.

(25) Levy, J. I., J. J. Buonocore and K. von Stackelberg (2010). "Evaluation of the public health impacts of traffic congestion: a health risk assessment." Environmental Health **9**(1): 65.

(26) Liu, H., X. Xu, M. O. Rodgers, Y. A. Xu and R. L. Guensler (2017). "MOVES-Matrix and distributed computing for microscale line source dispersion analysis." Journal of the Air & Waste Management Association **67**(7): 763-775.

(27) Lv, Y., Y. Duan, W. Kang, Z. Li and F. Y. Wang (2015). "Traffic Flow Prediction With Big Data: A Deep Learning Approach." IEEE Transactions on Intelligent Transportation Systems **16**(2): 865-873.

(28) Maroto, J., E. Delso, J. Felez and J. M. Cabanellas (2006). "Real-Time Traffic Simulation With a Microscopic Model." IEEE Transactions on Intelligent Transportation Systems **7**(4): 513-527.

(29) MnDOT (n.d.). MnDOT Connected Corridor Project Overview. www.dot.state.mn.us, MnDOT

(30) Mohd Zulkefli, M. A., P. Mukherjee, Y. Shao and Z. Sun (2016). "Evaluating Connected Vehicles and Their Applications." Mechanical Engineering Magazine Select Articles **138**(12): S12-S17.

(31) NYCDOT. (2019). "NYC Connected Vehicle Project." Retrieved 29 July, 2019, from https://www.cvp.nyc/.

(32) Oehmcke, S., O. Zielinski and O. Kramer (2016). kNN ensembles with penalized DTW for multivariate time series imputation. 2016 International Joint Conference on Neural Networks (IJCNN).

(33) OpenStreetMaps. (n.d.). "OpenStreetMap." Retrieved Dec 20 2019, 2019, from https://www.openstreetmap.org/.

(34) Pell, A., A. Meingast and O. Schauer (2017). Trends in Real-time Traffic Simulation.

(35) Pell, A., A. Meingast and O. Schauer (2017). "Trends in Real-time Traffic Simulation." Transportation Research Procedia **25**: 1477-1484.

(36) PennDOT. (2018, 2018). "Smart Corridor Initiatives." Transform76.com Retrieved July 29, 2019, from http://transform76.com/smart-corridor-initiatives/.

(37) Plotly. (2019, 2019). "Plotly JavaScript Graphing Library." Retrieved Dec 20 2019, 2019, from https://plot.ly/javascript/.

(38) Pop, M.-D. and O. Proștean (2018). "A Comparison Between Smart City Approaches in Road Traffic Management." Procedia - Social and Behavioral Sciences **238**: 29-36.

(39) RenewAtlantaBond. (2018, May 22, 2018). "RENEWATLANTATSPLOST." Retrieved 24 May, 2018, from http://renewatlantabond.com/.

(40) Roh, H.-J., Canadá and S. Sharma (2016). IMPUTATION OF MISSING CLASSIFIED TRAFFIC DATA DURING WINTER SEASON.

(41) Rouse, M. (2017, July 2017). "IOT Agenda." Smart City Retrieved 16 May, 2018, from https://internetofthingsagenda.techtarget.com/definition/smart-city.

(42) Saroj, A., S. Roy, R. Fujimoto, A. Guin and M. Hunter (2018). Smart City Real-Time Data-Driven Transportation Simulation Winter Simulation Conference, Gothenburg, Sweden.

(43) Seymour, T. (2018, 17 September 2018). "A2M2 connected corridor offers real-time, personalised travel information." Fleet Industry News Retrieved 29 July, 2019, from https://www.fleetnews.co.uk/news/fleet-industry-news/2018/09/17/a2m2-connected-corridor-offers-real-time-personalised-travel-information.

(44) Spławińska, M. (2015). "The Problem Of Imputation Of The Missing Data From The Continuous Counts Of Road Traffic." Archives of Civil Engineering **61**.

(45) Sturari, M., L. Catani, A. Mancini and E. Frontoni (2016). An integrated mobility system using real-time data for traffic simulation. 2016 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA).

(46) Sun, B., L. Ma, W. Cheng, W. Wen, P. Goswami and G. Bai (2017). An improved k-nearest neighbours method for traffic time series imputation. 2017 Chinese Automation Congress (CAC).

(47) Tak, S., S. Woo and H. Yeo (2016). "Data-Driven Imputation Method for Traffic Data in Sectional Units of Road Links." IEEE Transactions on Intelligent Transportation Systems **17**(6): 1762-1771.

(48) TDOT. (n.d.). "Interstate 24 Smart Corridor: Davidson and Rutherford Counties." Retrieved 29 July, 2019, from https://www.tn.gov/tdot/projects/region-3/i-24-smart-corridor.html.

(49) U.S. DOT, F. H. A. (2016). "New Approaches for Testing Connected Highway and Vehicle Systems." Retrieved 6 March, 2019, from https://highways.dot.gov/new-approaches-testing-connected-highway-and-vehicle-systems.

(50) U.S. DOT, F. H. A. (2017, July 2017). "In-the-Loop Simulations Provide Improved Methods for Testing of Connected Vehicle Technologies." Retrieved 6 March, 2019, from https://www.fhwa.dot.gov/publications/research/ear/17044/index.cfm.

(51) U.S. DOT, I. T. S. J. P. O. (2014). AERIS—Applications for the Environment: Real-Time Information Synthesis, Eco-Signal Operations Modeling Report**:** 428.

(52) U.S. DOT, I. T. S. J. P. O. (2014). The Smart/Connected City and Its Implications for Connected Transportation, John A. Volpe National Transportation Systems Center, U.S. DOT, Intelligent Transportation Systems Joint Program Office.

(53) U.S. DOT, I. T. S. J. P. O. (2017). Connected Vehicle - Safety, ITS Benefits, Costs, and Lessons Learned: 2017 Update Report.

(54) U.S. DOT, I. T. S. J. P. O. (n.d.). Smart City Challenge Lessons Learned. www.transportation.gov.

(55) U.S. DOT, I. T. S. J. P. O. (n.d.). "What Are Connected Vehicles and Why Do We Need Them?" Connected Vehicle Basics Retrieved 16 May, 2018, from https://www.its.dot.gov/cv_basics/cv_basics_what.htm.

(56) UCBerkeley. (n.d.). "Connected Corridors Program." Retrieved 29 July, 2019, from https://connected-corridors.berkeley.edu/.

(57) USDOT. (n.d.). "Connected Vehicle Pilot Deployment Program." Connected Vehicles Retrieved 29 July, 2019, from https://www.its.dot.gov/pilots/index.htm.

(58) VTTI. (n.d.). "Connected Vehicle Environment." Retrieved 29 July, 2019, from https://www.vtti.vt.edu/vcc/communications.html.

(59) Williams, B. M. (2001). "Multivariate Vehicular Traffic Flow Prediction: Evaluation of ARIMAX Modeling." Transportation Research Record **1776**(1): 194-200.

(60) Williams, H. (2018, 10 September 2018). "What is a smart city? How to define a smart city?" Retrieved 29 July, 2019, from https://www.computerworld.com.au/article/646458/what-smart-city-how-define-smart-city/.

(61) Wismans, L., E. de Romph, K. Friso and K. Zantema (2014). "Real Time Traffic Models, Decision Support for Traffic Management." Procedia Environmental Sciences **22**: 220-235.

(62) WYDOT. (2017). "WYOMING DOT Connected Vehicle Pilot." Retrieved 29 July, 2019, from https://wydotcvp.wyoroad.info/.

(63) Xiaowen, D., M. A. Ferman and R. P. Roesser (2003). A simulation evaluation of a real-time traffic information system using probe vehicles. Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems.

(64) Zhong, M. and S. Sharma (2006). "Matching Hourly, Daily, and Monthly Traffic Patterns to Estimate Missing Volume Data." Transportation Research Record **1957**(1): 32-42.

(65) Zhong, M., S. Sharma and Z. Liu (2005). "Assessing Robustness of Imputation Models Based on Data from Different Jurisdictions: Examples of Alberta and Saskatchewan, Canada." Transportation Research Record **1917**(1): 116-126.

(66) Zhuang, Y., R. Ke and Y. Wang (2019). "Innovative method for traffic data imputation based on convolutional neural network." IET Intelligent Transport Systems **13**(4): 605-613.

APPENDIX A: SIDEWALK, RAMP, AND CURB CUT INVENTORY AND CONDITION ASSESSMENT

1. INTRODUCTION

Providing safe and accessible pedestrian routes is integral to ensuring equal access, and can also encourage residents and employees to choose the healthy and sustainable choice of walking as their mode of transportation. Ensuring that pedestrian pathways comply with design guidelines adopted under provisions of the Americans with Disabilities Act (ADA) help to facilitate pedestrian accessibility and guarantees that all citizens, regardless of physical and cognitive ability, have a superior walking experience.

Developing a pedestrian-friendly and ADA-compliant pedestrian environment begins with the generation of a sidewalk link-and-node network for use in an asset management system. Asset management systems assist agencies in implementing sidewalk improvements by establishing a consistent method of identifying problems, quantifying impacts, and prioritizing sidewalk repair and maintenance. After the network is prepared, pedestrian infrastructure elements are inventoried (e.g., sidewalks, ramps, and curb cuts), for tracking in the asset management system. Then, the applicable features of each element (e.g., length, width, slope, cross-slope, surface condition, etc.) are inspected to ensure that the element is in conformance with ADA design guidelines. Deviations of any features from the design standards results in an ADA non-conformance finding, which must be identified and remedied within an ADA Transition Plan.

As part of the Renew Atlanta study conducted by the Georgia Institute of Technology for Renew Atlanta, researchers at the Georgia Institute of Technology developed the sidewalk network for four major corridors (see Figure 42) and inventoried the sidewalk elements of interest on each of the four corridors:

- Monroe Drive (Armor Drive to Decatur Street)
- Decatur Street (Boulevard to East Lake MARTA)
- Campbellton Road (Greenbriar Mall to Oakland City MARTA)
- North Avenue (Marietta Street to Candler Park Drive)

Once the inventory was complete, the research team employed semi-automated tools developed by the Georgia Tech team to assess the conditions of sidewalks, paths, and ramps for conformance with ADA design standards. In field studies, undergraduate research assistants deploy the Sidewalk Sentry™ system, which records geo-tagged video and vibration data from a wheelchair-mounted tablet. Field teams then use the Sidewalk Scout™ smartphone app to inventory the elements, enter field measurement data for ramp and curb cut features, and automatically assess compliance with ADA design standards.

The research team prepared four reports, one for each of the four corridors summarizing the research efforts. Each report first outlines federal accessibility design standards and guidelines as they apply to the transportation network (these standards define the parameters that are used by the asset management tools to assess ADA compliance of sidewalks and curb ramps). Subsequent chapters of ach report provide the methodology for generating sidewalk network data, followed by descriptions of how the Sidewalk Sentry™ and Sidewalk Scout™ systems are used in the field to record facility design issues, such as uneven sidewalk surfaces, potholes, and defective or missing curb ramps. Using the data entered by field inspectors, these systems automatically generate sidewalk, curb ramp, and curb cut inspection reports for each element (sidewalk sub-area, ramp, and curb cut), which are provided as report Appendices. The final analytical section of each report also provides a scoping estimate of sidewalk, ramp, and curb cut repair costs. The sidewalk network, analytical framework, and inspection results can assist the city in prioritizing sidewalk improvements.

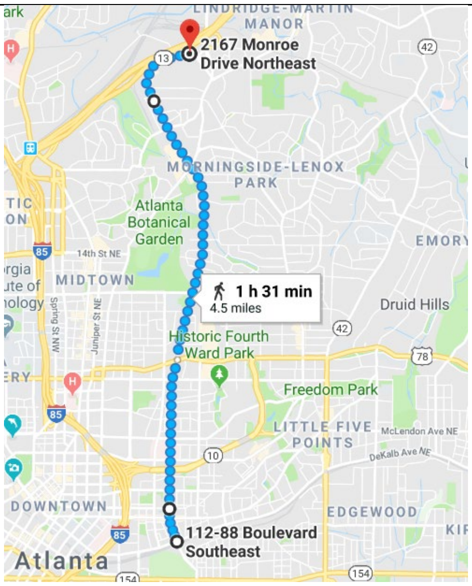
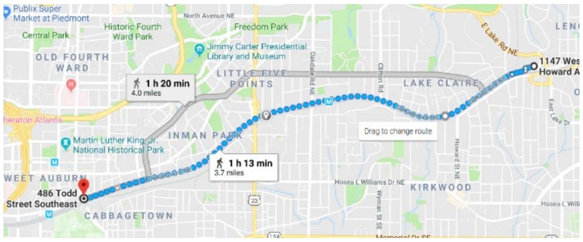
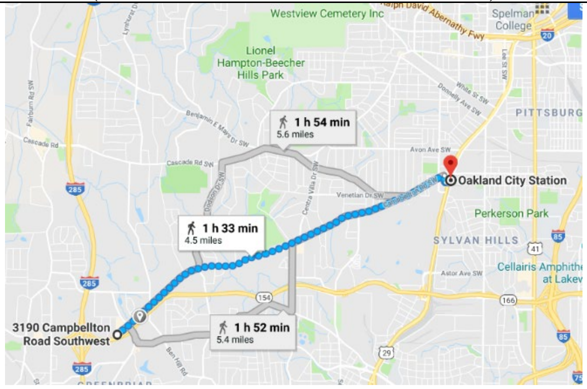
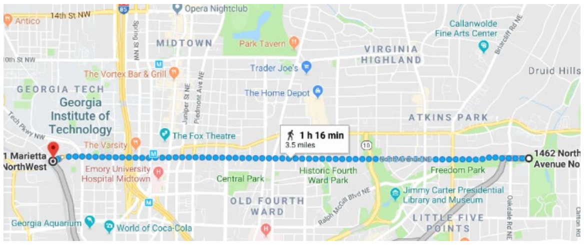
 2167 Monroe Drive Northeast MORNINGSIDE-LENOX PARK Atlanta Botanical Garden MIDTOWN 1 h 31 min 4.5 miles Historic Fourth Ward Park Freedom Park LITTLE FIVE POINTS Druid Hills DOWNTOWN EDGEWOOD 112-88 Boulevard Southeast Atlanta	 OLD FOURTH WARD Freedom Park 1 h 20 min 4.0 miles INMAN PARK LAKE CLAIRE 1147 Wes Howard A Drag to change route 1 h 13 min 3.7 miles 486 Todd Street Southeast CABBAGETOWN KIRKWOOD
Monroe Drive (Armor Drive to Decatur Street)	Decatur Street (Boulevard to East Lake MARTA)
 Westview Cemetery Inc Spelman College Lionel Hampton-Beecher Hills Park 1 h 54 min 5.6 miles PITTSBURG Oakland City Station Perkerson Park 1 h 33 min 4.5 miles SYLVAN HILLS 3190 Campbellton Road Southwest 1 h 52 min 5.4 miles	 MIDTOWN VIRGINIA HIGHLAND GEORGIA TECH Georgia Institute of Technology ATKINS PARK 1 h 16 min 3.5 miles 1 Marietta NorthWest 1462 North Avenue No Historic Fourth Ward Park Freedom Park OLD FOURTH WARD LITTLE FIVE POINTS
Campbellton Road (Greenbriar Mall to Oakland City MARTA)	North Avenue (Marietta Street to Candler Park Drive)

Figure 42: Four Corridors Selected for Atlanta Sidewalk Assessment

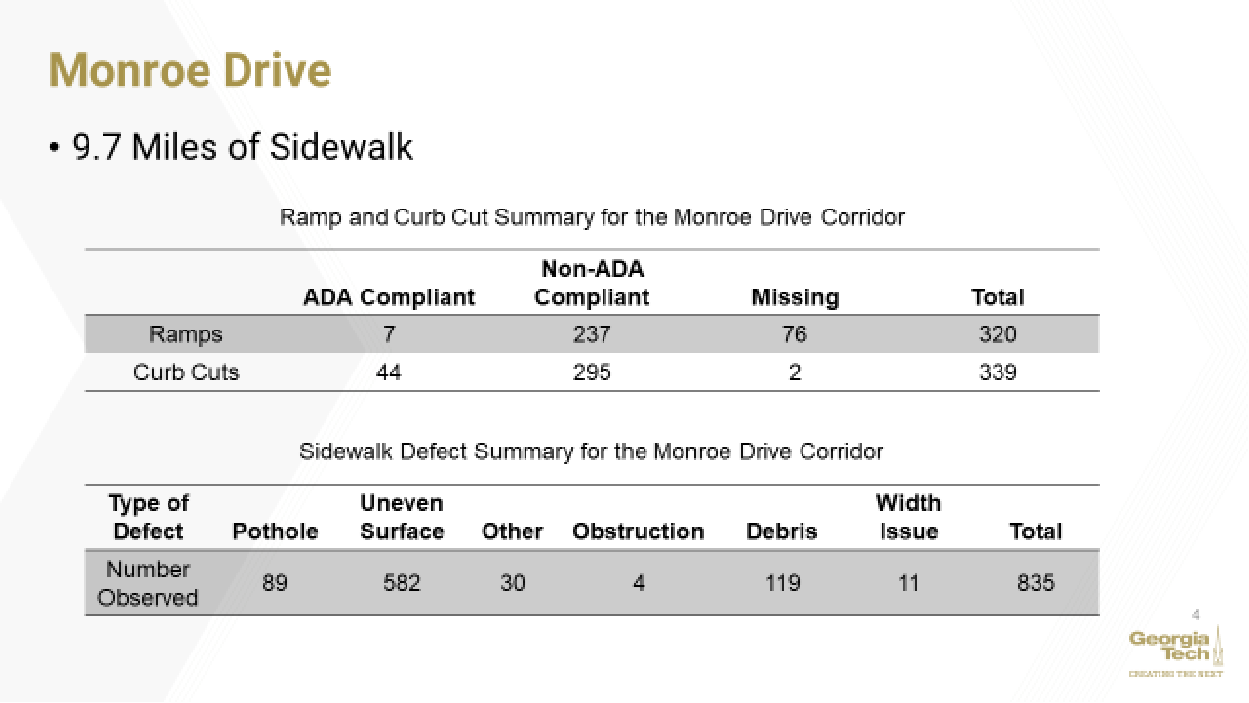
As noted above, the appendices to each report include an individual inspection reports for each ramp, curb cut, and sidewalk defects. The Appendices number in the hundreds of pages per report (i.e., one page per element inspection). Each one-page report describes the ADA-compliance issues recorded in the field, and maps the inspection results for use in geographic information systems.

2. SUMMARY OF SIDEWALK, RAMP, AND CURB CUT ASSESSMENTS

The summary of inspection results for the corridors is as follows:

2.1. Monroe Drive (Armor Drive to Decatur Street)

The research team inspected 9.7 miles of sidewalk, 320 ramps, and 339 curb cuts in the Monroe Drive study area between Armor Drive and Decatur Street. Inspectors identified a total of 835 sidewalk defects, 237 non-ADA-compliant ramps, 76 pedestrian crossing locations with missing ramps, and 295 non-ADA-compliant curb cuts. The basic costs for repair and construction of missing elements (excluding any extraordinary costs for moving power poles, storm sewer features, etc.) are estimated to be about $1.37 million.



Monroe Drive

- 9.7 Miles of Sidewalk

Ramp and Curb Cut Summary for the Monroe Drive Corridor

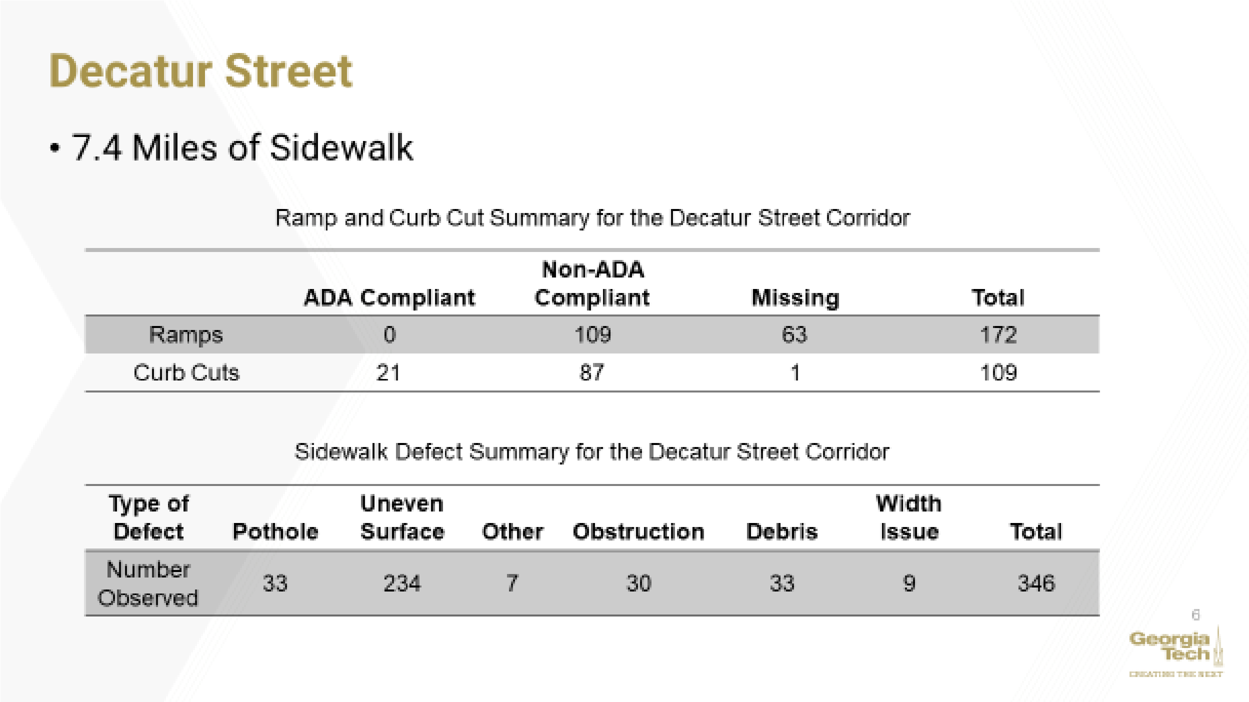
	ADA Compliant	Non-ADA Compliant	Missing	Total
Ramps	7	237	76	320
Curb Cuts	44	295	2	339

Sidewalk Defect Summary for the Monroe Drive Corridor

Type of Defect	Pothole	Uneven Surface	Other	Obstruction	Debris	Width Issue	Total
Number Observed	89	582	30	4	119	11	835

2.2. Decatur Street (Boulevard to East Lake MARTA)

The research team inspected 7.4 miles of sidewalk, 172 ramps, and 109 curb cuts in the Decatur Street (DeKalb Avenue) study area between Boulevard and East Lake MARTA. Inspectors identified a total of 346 sidewalk defects, 109 non-ADA-compliant ramps, 63 pedestrian crossing locations with missing ramps, and 87 non-ADA-compliant curb cuts. The basic costs for repair and construction of missing elements (excluding any extraordinary costs for moving power poles, storm sewer features, etc.) are estimated to be about $430,000.



Decatur Street

- 7.4 Miles of Sidewalk

Ramp and Curb Cut Summary for the Decatur Street Corridor

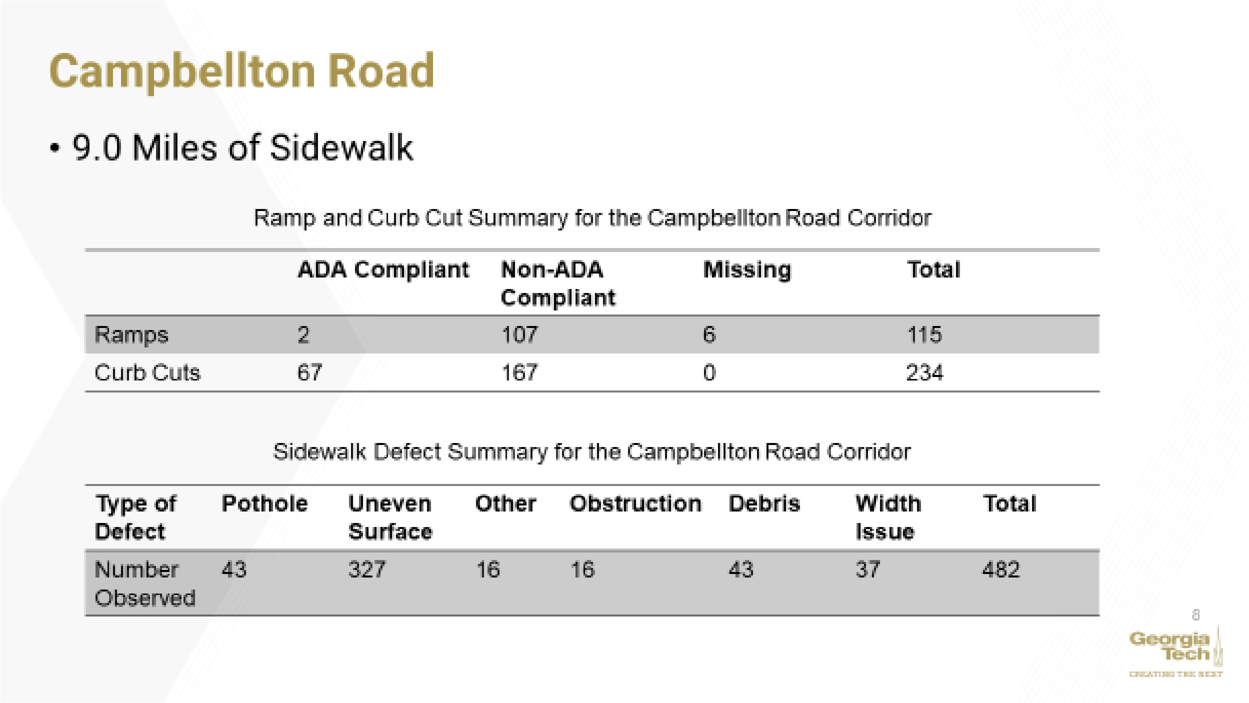
	ADA Compliant	Non-ADA Compliant	Missing	Total
Ramps	0	109	63	172
Curb Cuts	21	87	1	109

Sidewalk Defect Summary for the Decatur Street Corridor

Type of Defect	Pothole	Uneven Surface	Other	Obstruction	Debris	Width Issue	Total
Number Observed	33	234	7	30	33	9	346

2.3. Campbellton Road (Greenbriar Mall to Oakland City MARTA)

The research team inspected 9.0 miles of sidewalk, 115 ramps, and 234 curb cuts in the Campbellton Road study area between Greenbriar Mall and the Oakland City MARTA. Inspectors identified a total of 482 sidewalk defects, 107 non-ADA-compliant ramps, 6 pedestrian crossing locations with missing ramps, and 167 non-ADA-compliant curb cuts. The basic costs for repair and construction of missing elements (excluding any extraordinary costs for moving power poles, storm sewer features, etc.) are estimated to be about $740,000.



Campbellton Road

- 9.0 Miles of Sidewalk

Ramp and Curb Cut Summary for the Campbellton Road Corridor

	ADA Compliant	Non-ADA Compliant	Missing	Total
Ramps	2	107	6	115
Curb Cuts	67	167	0	234

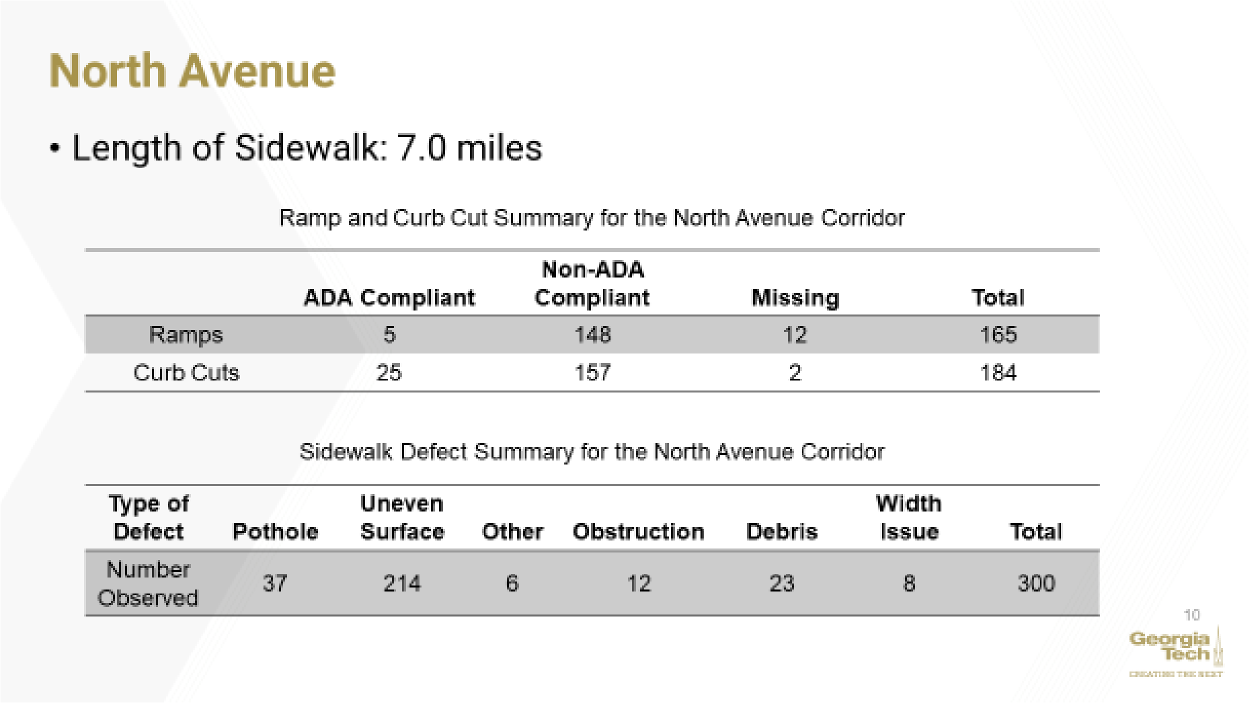
Sidewalk Defect Summary for the Campbellton Road Corridor

Type of Defect	Pothole	Uneven Surface	Other	Obstruction	Debris	Width Issue	Total
Number Observed	43	327	16	16	43	37	482

8

2.4. North Avenue (Marietta Street to Candler Park Drive)

The research team inspected 7.0 miles of sidewalk, 165 ramps, and 184 curb cuts in the North Avenue study area between Marietta Street and Candler Park Drive. Inspectors identified a total of 300 sidewalk defects, 148 non-ADA-compliant ramps, 12 pedestrian crossing locations with missing ramps, and 157 non-ADA-compliant curb cuts. The basic costs for repair and construction of missing elements (excluding any extraordinary costs for moving power poles, storm sewer features, etc.) are estimated to be about $570,000.



North Avenue

- Length of Sidewalk: 7.0 miles

Ramp and Curb Cut Summary for the North Avenue Corridor

	ADA Compliant	Non-ADA Compliant	Missing	Total
Ramps	5	148	12	165
Curb Cuts	25	157	2	184

Sidewalk Defect Summary for the North Avenue Corridor

Type of Defect	Pothole	Uneven Surface	Other	Obstruction	Debris	Width Issue	Total
Number Observed	37	214	6	12	23	8	300

3. REFERENCES

1. Boyer, D., D. Patel, D. Oswald, and R. Guensler (2019). Sidewalk, Ramp, and Curb Cut Inventory and Condition Assessment: North Avenue, Atlanta, GA. Georgia Institute of Technology, School of Civil and Environmental Engineering, Atlanta, GA. Prepared for Renew Atlanta. May 21, 2019.
2. Boyer, D., D. Patel, D. Oswald, and R. Guensler (2019). Sidewalk, Ramp, and Curb Cut Inventory and Condition Assessment: Campbellton Road, Atlanta, GA. Georgia Institute of Technology, School of Civil and Environmental Engineering, Atlanta, GA. Prepared for Renew Atlanta. April 30, 2019.
3. Boyer, D., D. Patel, D. Oswald, and R. Guensler (2019). Sidewalk, Ramp, and Curb Cut Inventory and Condition Assessment: Decatur Street (DeKalb Avenue), Atlanta, GA. Georgia Institute of Technology, School of Civil and Environmental Engineering, Atlanta, GA. Prepared for Renew Atlanta. March 7, 2019.
4. Boyer, D., D. Patel, D. Oswald, and R. Guensler (2018). Sidewalk, Ramp, and Curb Cut Inventory and Condition Assessment: Monroe Drive, Atlanta, GA. Georgia Institute of Technology, School of Civil and Environmental Engineering, Atlanta, GA. Prepared for Renew Atlanta. December 2018.